1

THE *DISASTER*™ SCHEDULING SYSTEM: A REVIEW AND CASE ANALYSIS

THESIS

Jefferson L. Severs, Captain, USAF

AFIT/GLM/LSM/91S-56

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GLM/LSM/91S-56

THE *DISASTER*™ SCHEDULING SYSTEM:
A REVIEW AND CASE ANALYSIS

THESIS

Jefferson L. Severs, Captain, USAF

AFIT/GLM/LSM/91S-56

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GLM/LSM/91S-56

THE *DISASTER*™ SCHEDULING SYSTEM: A REVIEW AND CASE ANALYSIS

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Logistics Management

Jefferson L. Severs, B.B.A.

Captain, USAF

September 1991

Acknowledgements

I would like to express my sincere appreciation to those people who helped make this research possible. First, Lieutenant Colonel Richard Moore was instrumental in helping me select the topic and in keeping me "on track" throughout the research. Next, I am thankful to Sacramento Air Logistics Center, and in particular Ms. Eva Ugarkovich, for funding my travel to San Jose to perform the case analysis. Without this funding, a real-world examination of a *DISASTER*™ application would likely not have been possible. Another important contributor to this research was the Zycon Corporation, in particular Mr. Larry Shoemaker. The patience and support of Mr. Shoemaker and other involved Zycon managers contributed greatly to this research effort. Finally, and most importantly, I wish to thank my family, without whom the entire project would not have been possible. My wife, Gay, and children, Hugh and Brooke, were unselfish, supportive, and understanding despite the countless hours I declined to join them in family activities. Thank you!

Jeff L. Severs

Table of Contents

Page

List of Figures

AFIT/GLM/LSM/91S-56

Abstract

This research examined the capabilities of a new scheduling system, called *DISASTER™*, based on the theory of constraints. Use of this system may be beneficial to the Department of Defense, particularly Air Force Logistics Command maintenance organizations. The research begins with a review of literature pertaining to manufacturing planning and control systems, including material requirements planning (MRP), manufacturing resource planning (MRP II), AFLC's Depot Maintenance Management Information System (DMMIS), just-in-time (JIT) manufacturing, and the theory of constraints (drum-buffer-rope scheduling). The research then examined the logic and operation of the *DISASTER™* system. Next, the research examined the implementation of *DISASTER™* at a commercial printed circuit board manufacturer.

The research indicates that a growing number of experts now believe that MRP systems do not provide adequate shop-floor scheduling and control. While AFLC's DMMIS will certainly be an improvement over the command's present method of operation (i.e., 1950s and 1960s vintage batch processing systems), this research suggests that other more recently developed alternatives, such as *DISASTER™*, may be even more advantageous. Furthermore, the research indicates that AFLC should have all the necessary requirements for implementing *DISASTER™*. The most significant obstacle appears to be the need for a a significant change in management philosophy, based on the theory of constraints.

THE DISASTER™ SCHEDULING SYSTEM: A REVIEW AND CASE ANALYSIS

I. *Introduction*

General Issue

Due to the United States' current fiscal problems and the widespread perception of a reduced Soviet threat, the Department of Defense is facing significant manpower and funding reductions. Many Air Force organizations, including Air Force Logistics Command (AFLC), are now investigating ways to reduce cost and improve productivity. Despite recent successes in Operation Desert Storm, little debate exists concerning the productivity improvement potential for AFLC's industrially-funded maintenance operations.

AFLC/MA's current approach to improve productivity is implementation of a system called the Depot Maintenance Management Information System (DMMIS). The system is currently being developed using a modified manufacturing resource planning system (MRP II), a commercial off-the-shelf software package. Subsequent to the decision to base DMMIS on MRP II logic, a new manufacturing philosophy based on Dr. Eliyahu Goldratt's *General Theory of Constraints* (TOC) has gained widespread popularity. Goldratt recently released a microcomputer-based scheduling system, called *DISASTER™*, that is based on TOC principles. AFLC has just begun to investigate TOC for application to depot maintenance planning and control.

Specific Problem Statement

Despite the tremendous growth in popularity of MRP II, there is growing concern among operations management experts today that materials requirements planning (MRP)-based approaches have failed to deliver the promised results in manufacturing (Kanet, 1988:57). Implementation of MRP systems is very expensive and more often than not fails to produce the advertised benefits (Kanet, 1988:57; Chase and Aquilano, 1989:656). For the AFLC manufacturing environment, in particular, there is concern that the MRP II-based DMMIS system will not provide appropriate scheduling and control. Goldratt's new *DISASTER™* software may offer an alternative in AFLC's search for a reliable scheduling and control system.

Investigative Questions

1. What is the basic premise behind materials requirements planning (MRP)?

2. What is manufacturing resource planning (MRP II) and how does it differ from MRP?

3. What is the planned approach for DMMIS?

4. What are potential problems and limitations of MRP-based systems?

5. What TOC concepts form the basis of the *DISASTER™* scheduling system?

6. What are the specific characteristics of the *DISASTER™* system and how does this system differ from conventional scheduling approaches?

7. What are the requirements for and the potential problems/obstacles inherent with implementation of the *DISASTER™* system?

Scope

DISASTER™ is a software package intended to provide a complete production management information system. The software is being developed and released in three separate phases: phase one, released in February 1991, includes the scheduling capability; phase two incorporates functions for control; and phase three provides the capability to perform "what-if" analyses.

The primary objective of this research is to investigate the phase-one scheduling software. Before one can properly analyze *DISASTER*™, he must understand manufacturing planning and control systems in general. To accomplish this objective, this research first reviews literature pertaining to materials requirements planning (MRP), manufacturing resource planning (MRP II), AFLC's Depot Maintenance Management Information System, and (briefly) just-in-time manufacturing. In addition, the research provides an in-depth review of principles of the theory of constraints (TOC), in particular drum-buffer-rope, that provide the basis for the *DISASTER*™ program.

Once these areas have been sufficiently investigated, the research will then focus on the software package itself. The purpose of reviewing the software is not to provide detailed instructions for its use--these are contained in its accompanying documentation and manuals. The intent is to examine conceptually how the program works, to discuss its capabilities, and to identify potential benefits accompanying its use.

Finally, the research will include a single, holistic case analysis of a company that is currently implementing the *DISASTER*™ information system. As noted, the *DISASTER*™ software was only recently

released (February 1991). To date, although several companies are serving as development partners, no one is using this system to schedule manufacturing operations; however, one company, the Zycon Corporation, appears to be very close to releasing the program to the shop floor. Analysis of Zycon's efforts represents a unique, first-time opportunity to investigate implementation of the software. Inclusion of this case analysis to supplement the literature review and software analysis will provide a level of external validity that would not be possible with the literature review and software analysis alone.

Background

The General Theory of Constraints and DISASTER™. Dr. Eliyahu Goldratt first gained widespread recognition after the publication of his best-selling book, *The Goal*. This book presented Goldratt's ideas about manufacturing technology in an entertaining, easily understood novel. Goldratt's first attempt to market a computer-based manufacturing scheduling and control system was his Optimized Production Technology (OPT) program. Basically, OPT was a two-part package: a mainframe computer program that simulated the manufacturing planning environment and a set of radical shop-floor management rules (Powell, 1984:100). The cost of implementing OPT by commercial businesses was prohibitive for all but the largest companies. In some companies it cost over $500,000; however, as of 1983, about 20 fortune 500 companies including Ford, Westinghouse, General Motors, General Electric, and RCA had experimented with it (Bylinsky, 1983:121). Since the release of *The Goal*, Goldratt has released a number of follow-on books (*The Race*, *The Theory of Constraints*, and *The Haystack Syndrome)*, and he recently released a microcomputer-based revision of his original OPT program,

called *DISASTER™*. Goldratt's philosophy is now commonly referred to as the *theory of constraints* or synchronized manufacturing.

Goldratt chose to name his new information system "disaster" for a very good reason: although it can be a very powerful tool, if used without the skills and knowledge necessary to control the power of the package, its use can lead to disaster (Avraham Y. Goldratt Institute, 1990d:1). The software's documentation package uses an analogy comparing *DISASTER™* to a surgeon's scalpel. Only if used by people with the appropriate knowledge and skills will the package be beneficial (Avraham Y. Goldratt Institute, 1990d:1). This idea is a key point--the more thoroughly the users understand the principles of TOC, the more benefits they may reap from its use, and only then will they be able to realize its full potential (Avraham Y. Goldratt Institute, 1990d:1).

Applicability to AFLC Maintenance Operations. Air Force Logistics Command's core functions of requirements, acquisition, maintenance, and distribution are very dependent upon accurate, reliable information systems. Currently, AFLC logistics processes are supported by approximately 500 data systems (AFLC, 1989a:5). Unfortunately, many of these systems date back to the 1950's and 1960's and have incorporated few technology improvements. AFLC established the Logistics Management Systems (LMS) modernization program to update all of the AFLC management information systems.

The maintenance portion of the effort is called the Depot Maintenance Management Information System (DMMIS). Currently, organic depot maintenance is supported by more than 50 individual information systems (AFLC, 1990b:10). DMMIS will replace 29 of these outdated batch systems (AFLC, 1990b:10). Like nearly all U.S. manufacturers with sales

over $10 million, AFLC decided to base DMMIS on modified commercial MRP II software (Chase and Aquilano, 1989:624). DMMIS is intended to improve AFLC's ability to forecast, plan, and control depot maintenance activities (ALFC, 1989:35). The system is also expected to achieve reductions in inventory and lead times (AFLC, 1989:35). DMMIS phased implementation began in FY90 with OO-ALC/MAN as the test site, with follow-on implementation planned for the remaining depots. Theses by Finnern (1988) and Faulkner (1989) reviewed AFLC implementation of DMMIS in depth.

II. Review of the Literature

Introduction

This chapter provides a review of literature pertaining to current manufacturing planning and control systems. This chapter reviews traditional manufacturing philosophies and how they relate to implementation of the Depot Maintenance Management Information System (DMMIS). In addition, this chapter provides a discussion of TOC concepts which must be understood to comprehend the processing logic of the *DISASTER*™ scheduling system. This chapter begins with an overview of materials requirements planning (MRP)-based planning and control systems. This section outlines the basic logic behind MRP systems, reviews manufacturing resource planning (MRP II), describes planned DMMIS operations, and identifies potential problems with MRP-based systems. Next, due to the similarities between just-in-time (JIT) manufacturing and TOC, JIT will be briefly introduced. Finally, this chapter provides a more in-depth review of relevant TOC concepts.

Materials Requirements Planning (MRP)

Following World War II, waste was not a principal concern of American manufacturers. Their primary concern was meeting the existing high level of demand, so they maintained very large stockpiles of inventory to ensure there were no disruptions in production (Umble and Srikanth, 1990:7). The traditional American model for manufacturing still assumes that it is efficient to have large manufacturing scales with buffer stocks to ensure that all machines and workers are continually operating (Cusumano, 1988:32). In addition, traditional manufacturing approaches stress high levels of worker and equipment

specialization, extensive automation, and long production runs on large machines that require long setup times (Cusumano, 1988:32). Many U.S. firms also employ large numbers of inspectors using statistical sampling to ensure production quality is acceptable (Drucker, 1990:96). One of the most important aspects of this traditional system is the use of economic order quantity theory as the basis for establishing what lot (batch) sizes for manufacturing will result in the lowest cost of production (Fox, 1984:59). Using the economic order quantity formula, American manufacturers attempt to balance holding and setup costs. What often results is an attempt by managers to maximize production runs at the expense of smooth flow and improved customer responsiveness.

In contrast to newer manufacturing philosophies such as those employed in Japan, Western manufacturers place emphasis on speed of individual processing time, ("faster must be cheaper") and on protection or contingency in case of disruptions. This practice requires extra inventory, capacity, and manpower (Hay, 1988:16). The "push" concept is the basis for the traditional U.S. approach to production scheduling. Under a "push" scheduling system, each station delivers material or assemblies according to a master production schedule designed to run all machines and supply all materials regardless of any problems that might develop at downstream work stations (Cusumano, 1988:32). This method often results in excess inventory buildups, because when problems occur at one station, preceding operations continue to work at normal levels. Until the past decade, U.S. manufacturers did not view extra accumulations of inventory negatively. Plants simply used the excess inventory to "serve as a wall to shield the plant from disruptions in the production process" (Umble and Srikanth, 1990:16). American

managers have traditionally viewed (and indeed accounted for) inventory as an asset.

Materials requirements planning (MRP) is the most widely used U.S. approach to planning and scheduling manufacturing operations (Fox, 1984:60). Most U.S. manufacturing firms with sales greater than $10 million use a materials requirements planning system (Chase and Aquilano, 1989:624). MRP can be defined as a push scheduling system that uses bills of material, inventory data, and the master production schedule to calculate requirements for material (Umble and Srikanth, 1990:8;Cusumano, 1988:34).

The objectives of MRP are to control inventory levels, assign operating priorities for items, and plan capacity to load the production system. Basically, it involves getting "the right materials to the right place at the right time" (Chase and Aquilano, 1989:627). The major difference between MRP systems and their predecessors is the significant computerization of the process. In addition, the system facilitates much more detailed record keeping and attempts to improve lines of communication. MRP systems use extensive computer control to link inventory and scheduling systems, resulting in schedules that identify specific parts and materials required to produce end items, the exact numbers needed, and dates when orders for materials should be released to production (Chase and Aquilano, 1989:626).

As shown in Figure 1, the inputs to an MRP system are the bills of material, the master production schedule, and the inventory status file. Typical of push systems, a master production schedule dictates flow, directing deliveries of material and partially assembled components regardless of whether work stations are ready to receive them (Cusumano,

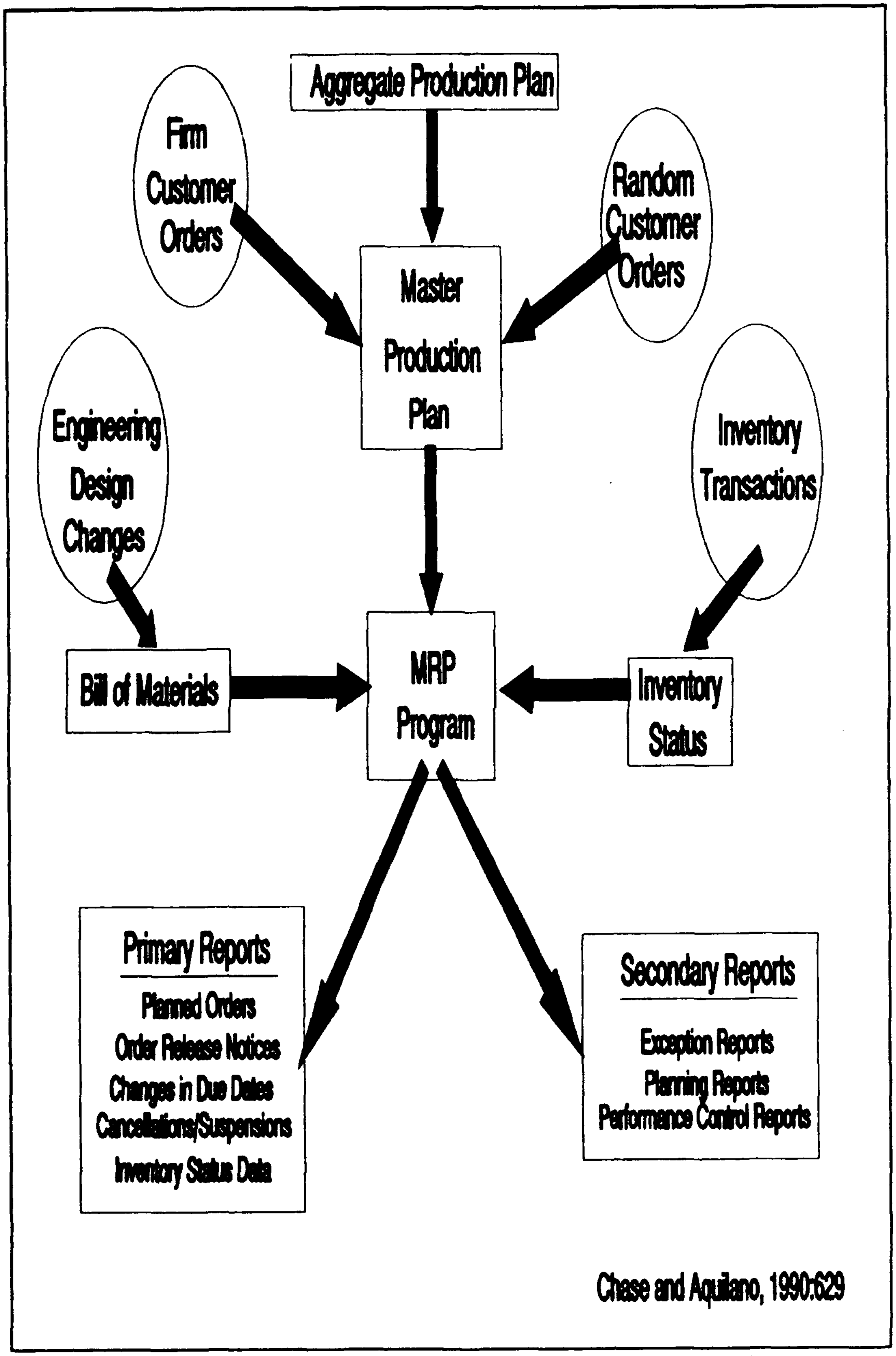

Figure 1. Standard MRP Inputs and Outputs

1988:34). The basis for MRP is computer-generated schedules that are tied directly to market demand, but which fail to adjust to the inevitable--changing shop or supplier conditions (Cusumano, 1988:36). The master production schedule provides the specific quantities of end-items required for the particular job. The MRP program then accesses on-hand and on-order information from the inventory status file and uses information from the bill of material file to "explode" requirements for all components and assemblies needed to produce the required end items.

Outputs from the MRP system include primary and optional reports such as those shown in Figure 1. For these outputs to be valid, the system must have accurate bills of material, a valid master production schedule, and detailed, accurate inventory records. A standard MRP system produces primary reports, relating to the quantities and timing of orders. Some MRP systems have been modified to produce optional reports, such as exception reports. For example, "net-change" MRP systems are activity driven--instead of re-exploding the entire material plan when unplanned activities occur, these systems generate exception reports that identify only the changes from the current plan (Chase and Aquilano, 1989:637). While such a system may seem worthwhile, in reality these systems often cause excessive "system nervousness." If net-change runs are performed too frequently, so many exceptions are identified that the productive system is unable to react properly (and often overreacts).

Manufacturing Resource Planning (MRP II)

Although originally intended only as a way to order material (hence *materials* requirements planning), MRP has now evolved to encompass a wide range of production functions. During the 1970's,

companies began to find it useful to integrate the capacity and material planning systems with order entry, purchasing, shop floor control, accounting, and other major data systems (Demmy and Giambrone, 1990:8). These "upgraded" MRP systems tied all of the separate manufacturing functions together and thus came to be known as manufacturing resource planning, or MRP II. MRP II is more than just a manufacturing material system--it attempts to link various departments within the entire manufacturing company using a single, integrated database (Metzger, 1984:52). This database includes information from all functional areas and allows all departments to access any information that might be beneficial in managing operations (Kilmer, 1986:20).

In addition to integration, MRP II systems provide a feedback mechanism between execution and planning stages that permits monitoring of production activity and facilitates decision making (Metzger, 1984:52-3). When a MRP system incorporates feedback, it is termed a *closed-loop* MRP. According to the American Production and Inventory Control Society (APICS) dictionary, a closed-loop system is defined as:

> A system built around materials requirements planning and also including the additional planning functions of production planning, master production scheduling, and capacity requirements planning. Further, once the planning phase is complete and the plans have been accepted as realistic and attainable, the execution functions come into play. These include shop-floor control functions of Input-Output measurement, detailed Scheduling and Dispatching, plus anticipated delay reports from both the shop and vendors, Purchasing Follow-Up and Control, etc. The term *closed-loop* implies that not only is each element included in the system, but also that there is feedback from the execution functions so that planning can be kept valid at all times. (APICS Dictionary, 1984:4)

The closed-loop characteristic is a key difference between MRP II and standard MRP. Feedback allows MRP II to identify when adjustment is required by determining discrepancies between planned materials and available capacity. When variances occur, changes must be made to

either the capacity (adding overtime, subcontracting, etc.) or the schedule (Demmy and Giambrone, 1990:8). In a sense, feedback enables MRP II to integrate long-range strategic planning with detailed execution and provides tools for quickly evaluating and modifying the plans (Demmy and Giambrone, 1990:8).

Expansion of MRP to include other manufacturing functions was expected. The central idea behind MRP II is that clear and precise communication throughout the organization is absolutely necessary for efficient manufacturing planning and control. The intent of new MRP II systems is 1) to monitor all functions as discussed above through a *closed-loop* system, and 2) to provide some capability to perform simulation of manufacturing processes (Chase and Aquilano, 1989:649). MRP II allows "everyone (purchasing, marketing, production, accounting) to work with the same plan, use the same numbers, and perform simulation to plan and test strategies" (Chase and Aquilano, 1989:649). Due to its additional capabilities, MRP II is now widely used as a way to schedule (a method to maintain valid due dates on orders) and to control shop-floor functions such as input-output measurement (Umble and Srikanth, 1990:8).

In the same manner as MRP systems, all MRP II systems use a common software algorithm that analyzes requirements for end items; however, as noted above, MRP II systems encompass more functions including shop floor control and capacity planning (Kilmer, 1986:20; Demmy and Giambrone, 1990:7). Figure 2 depicts the typical functions performed by a MRP II planning and control system. Vollman, Berry, and Whybark (1984) categorize the functions as *front-end*, *engine*, and *back-end* activities, based on the scope/scale of the functions.

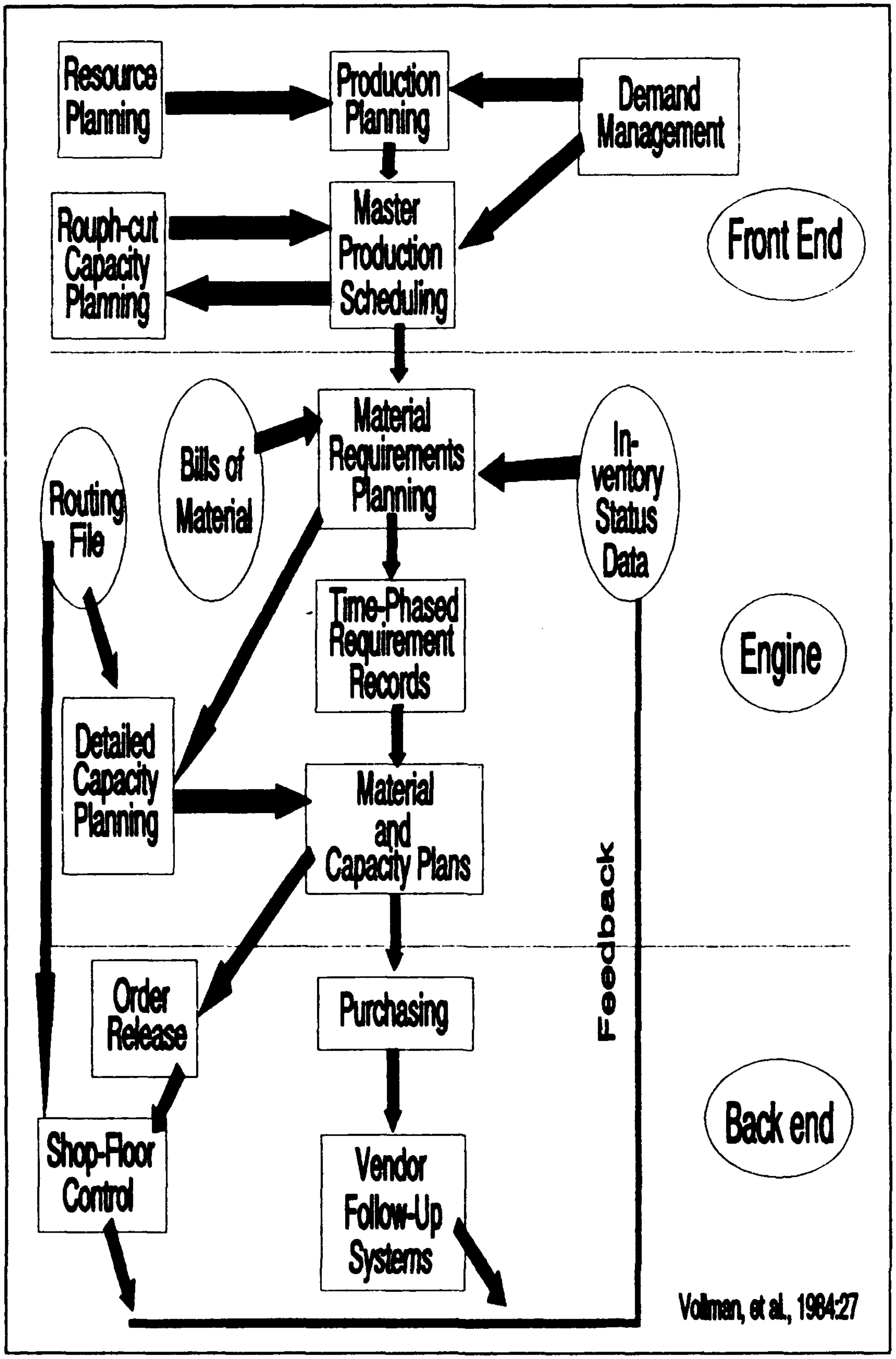

Figure 2. Manufacturing Planning and Control System

Front End. The "front end" is the set of activities that establishes overall company direction (Vollman, Berry and Whybark, 1984:12). Demand management involves both firm customer orders and forecasted, random end-item requirements. Demand for repair parts and supplies (items less complex than the end item) is normally input directly to the MRP module (Chase and Aquilano, 1989:629). The basic purpose of demand management is to "coordinate all business activities that place demands on the manufacturing activity" (Demmy and Giambrone, 1990:8). The objective of production planning is to determine requirements (normally by month or quarter) for broad product lines or groups. The resulting production plan normally includes some rough consideration of resource requirements and limitations. The master production schedule (MPS) disaggregates the production plan into requirements for specific end items, including some rough-cut capacity planning (RCCP) capability. RCCP reviews the MPS to ensure that no obvious capacity constraints exist (Chase and Aquilano, 1989:548).

Engine. The "engine" is the set of functions that involve detailed material and capacity planning (Vollman, Berry and Whybark, 1984:12). The MRP module takes the end-item requirements from the MPS and calculates specific time-phased requirements for components and subassemblies. The material plan drives all calculations for detailed capacity planning (Demmy and Giambrone, 1990:8). When changes occur to the MPS, the MRP program can perform an integrated computation that will update both the material plans and the shop floor schedules (Demmy and Giambrone, 1990:7). In addition to the MRP module, MRP II includes a module for capacity requirements planning (CRP). CRP "provides a detailed schedule of when each operation will be run on each work center

and how long it will take" (Chase and Aquilano, 1989:548-9). The objective of CRP is to compute the total labor and machine hours required to complete each operation for all open and planned orders, then summarize by work center and time period to obtain the total resources required to support all scheduled outputs (Demmy and Giambrone, 1990:7). The CRP module uses the routing files to determine how the work can be scheduled and loaded on the resources, and it is rerun periodically to provide accurate information needed for execution (Kilmer, 1986:20). MRP II programs that include a CRP module permit rescheduling to try to spread the load more evenly between work centers; however, this function is often not performed (Chase and Aquilano, 1989:647).

Back End. The "back end" of an MRP II system includes the execution functions. Purchasing control deals with the acquisition and control of purchased items, as specified by the materials plan. This function provides detailed planning data for vendor scheduling and releases/monitors purchase orders (Demmy and Giambrone, 1990:8). Shop floor control includes functions such as establishing priorities (sequencing and resequencing), dispatching, and reviewing and reporting on work status.

MRP II Logic Applied to AFLC Depot Maintenance

Air Force Logistics Command overhauls, repairs, or modifies 1,200 aircraft, 6,400 aircraft engines, 1.1 million reparable assemblies (communications and electronics systems, generators, landing gears, etc.) annually (Demmy and Giambrone, 1990:8-9). AFLC has six major industrial facilities and over 37,000 people involved in maintenance activities, and spends over $2.5 billion annually to support its

operations (Demmy and Giambrone, 1990:8). The success of the command's depot maintenance mission is highly dependent upon the availability of enormous amounts of timely and accurate information (AFLC, 1988:1). During the 1980's, AFLC used separate information systems for financial, production, and resource management. Current organic data systems include more than 50 individual systems that are primarily 1960's vintage sequential tape interface systems using batch processing (AFLC, 1990b:10). Unfortunately, this system fails to provide timely information required for AFLC maintenance operations (AFLC, 1988:2). The current planning environment requires labor-intensive and voluminous paperwork processing in a batch-mode environment, resulting in information often becoming obsolete even before it is received (AFLC, 1990b:10). The current system design is little more than "a primitive mechanization of concepts and procedures using 1950's and 60's generation computer equipment and processing methodologies" (AFLC, 1988:2).

During the 1980's, AFLC began to investigate alternatives for upgrading their logistics information systems. As a result of a 1985 study performed by a private consulting firm (Deloitte, Haskins, and Sells), AFLC decided to use commercially available MRP II software as the basis for upgrading their maintenance information systems (AFLC, 1988:2). AFLC plans to automate the maintenance system using an on-line, real-time system that is networked with other information systems (AFLC, 1990a:4). The new Depot Maintenance Management Information System (DMMIS) will replace 29 current batch and 4 on-line systems (AFLC, 1988:1).

DMMIS source selection began in 1986 and ended in 1988 with the selection of Grumann Data Systems (GDS) as the prime contractor (AFLC, 1989b:15). GDS proposed a four-phase implementation approach at all operational sites, with Ogden Air Logistics Center (ALC) as the first site. System operation will be concurrent with the existing system until all divisions within each ALC have transitioned to DMMIS (AFLC, 1989b:15). Phase I was the implementation of the exchangeables production system (EPS) at each of the ALC's (AFLC, 1990b:10). The EPS replaced 5 of the 29 existing systems and supports scheduling of exchangeables workloads, transaction processing, and material control (AFLC, 1990b:10). Phases II and III will incorporate the functions of EPS and replace the 24 remaining maintenance systems with the MRP II software (AFLC, 1990b:10). System hardware requirements include IBM 3090s as DMMIS central processors at each ALC and AGMC. Remote devices will be Z-248 work stations, CMI 6019 terminals, PTC701 hand-held bar terminals, bar code readers, and Gemicom or ALPS printers (AFLC, 1989b:15).

The objective of DMMIS is to provide AFLC maintenance organizations with the capability to effectively determine and assure that the necessary resources are available at the centers to perform their missions successfully (AFLC, 1988:1). DMMIS is intended to improve AFLC's ability to forecast, plan, and control depot maintenance activities. In addition, AFLC expects the system to reduce inventory and lead times. Typical of MRP II-based systems, DMMIS is intended to provide effective planning for *all* resources within AFLC maintenance organizations. The DMMIS software will encompass all of the traditional MRP II planning functions, such as business planning, demand

forecasting, production planning, material planning, capacity planning, master scheduling, and shop-floor control (AFLC, 1990a:7). In addition, the system includes functions necessary to plan workloads, establish and maintain production lines, schedule skills and parts support, and operate on an industrially funded basis (AFLC, 1989b:15). The system will address operational planning in units, financial planning in dollars, and will have a simulation capability to test "what-if" situations" (AFLC, 1990a:7). AFLC expects the closed-loop nature of MRP II to allow maintenance management to continually plan resources, execute schedules, and evaluate performance (AFLC, 1990b:10).

AFLC Maintenance versus Commercial Manufacturing. The majority of the characteristics of AFLC depot maintenance planning and control are the same as in large-scale manufacturing. Advance planning is required, detailed requirements and resource availability must be determined and monitored, and large numbers of orders have to be scheduled, released, and monitored (Demmy and Giambrone, 1990:9). In addition, most of the data elements (BOM file, inventory file, etc.) are still basically the same. Despite the similarities, there are some key differences that affect AFLC's system of manufacturing planning and control. Before identifying these differences, it is necessary to understand the typical functions involved in AFLC maintenance operations. As shown in Figure 3, when a reparable asset arrives, it is first disassembled into its major components, cleaned, and subjected to any necessary nondestructive inspection. Since each recovered component may have different processing requirements, a thorough evaluation and inspection must be performed to identify the precise processing that will be required for each component. During evaluation and inspection (E&I), each component

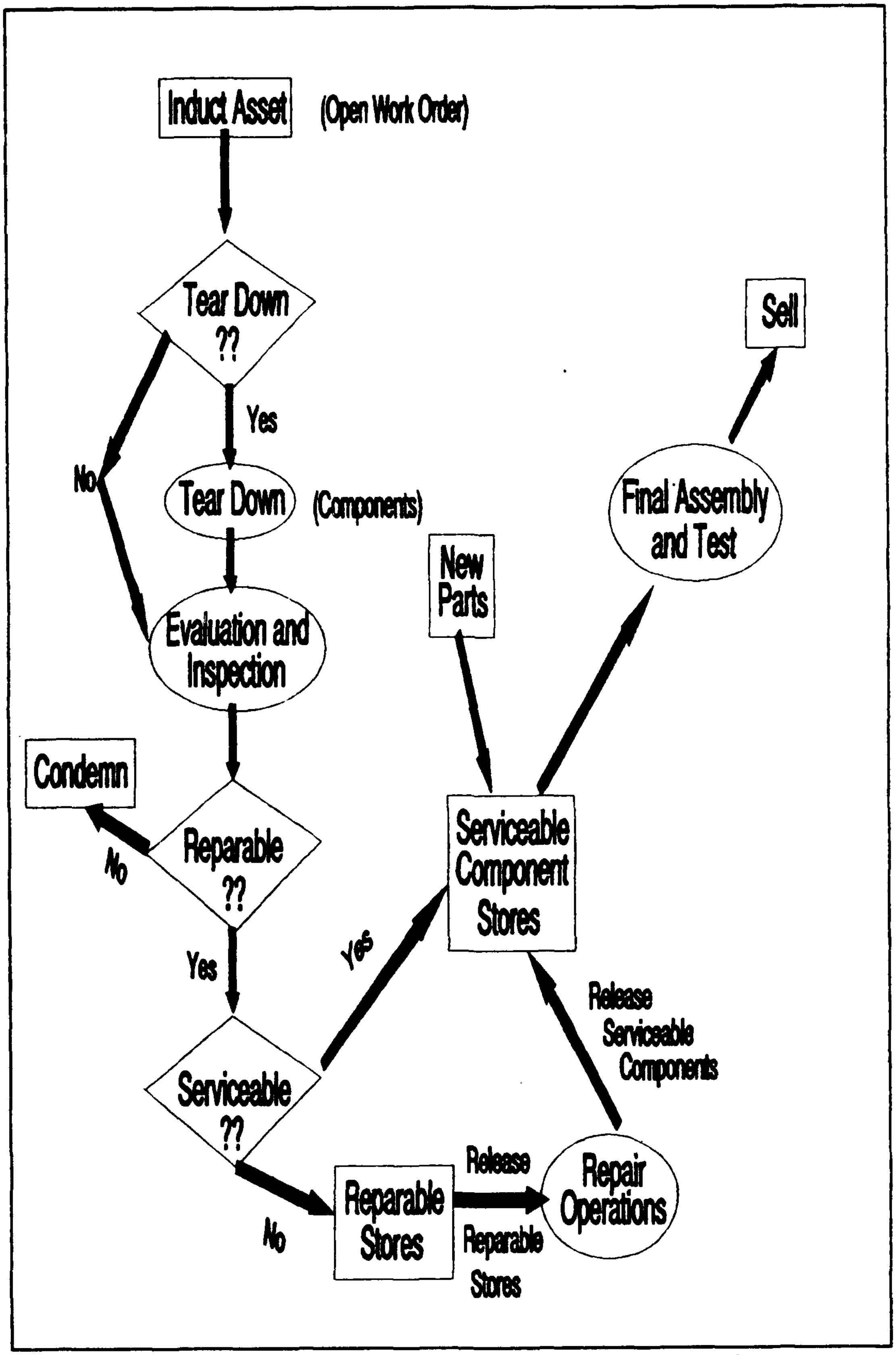

Figure 3. Typical Maintenance Repair Flow

is condemned (uneconomical or impossible to repair), sent to the serviceable inventory (no repair needed), or routed to the appropriate repair department. Once the components are repaired, they are sent to the serviceable inventory and, based on the need date for the weapon system, they are pulled from inventory, assembled, tested, and sold/shipped or returned to supply.

Clearly, a significant difference between normal manufacturing and maintenance is the high level of uncertainty involved with maintenance operations. In manufacturing, a modular BOM can be used to determine the exact material requirements; however, for repair operations, exact materials and required processes cannot be determined until after the E&I stage. Not only is there uncertainty with respect to what operations need to be performed, but also there is uncertainty regarding the availability of reparable core assets (Demmy and Giambrone, 1990:9). The success of maintenance planning is highly dependent upon accurate forecasting of the arrival of reparable assets at the repair facility.

In addition to the higher level of uncertainty, AFLC maintenance organizations also have a unique mission and they must consider repair as well as purchase for material acquisition decisions (Demmy and Giambrone, 1990:9). Unlike the traditional MRP II environment, where profit is the goal, AFLC's primary goal is readiness (AFLC, 1990b:i). AFLC's maintenance process is designed to provide effective, deployable, and economical procedures during peacetime operations and to provide the capability for rapid and effective surge capability during emergencies. The command must maintain the capability to respond quickly and appropriately in the event of war. While material requirements in commercial manufacturing firms are all filled through purchases,

maintenance managers must decide whether to purchase or repair items to satisfy the production requirements. Repair is usually cheaper than purchase, and often it is the only alternative.

Demmy and Giambrone (1990) also contend that AFLC maintenance organizations interface with only one customer, the Directorate of Distribution (DS), and one supplier, the Directorate of Materiel Management (MM); however, from a more macro level, DS has a number of different suppliers and MM has multiple customers. Any AFLC productive system must consider more than just the micro (MA) view--it must consider the entire productive system.

To account for some of the unique features of AFLC depot maintenance, portions of the standard commercial MRP II software are being modified or extended (Demmy and Giambrone, 1990:8). First, DMMIS will be modified to support the construction of routings tailored to particular reparable items (Demmy and Giambrone, 1990:11). Due to the unique E&I requirements inherent in repair operations, DMMIS will also enhance the ability to plan and track the disassembly of items (Demmy and Giambrone, 1990:11). DMMIS will also provide extensive data collection and analysis support for quality assurance, statistical process control, reliability control, and unique government accounting procedures (Demmy and Giambrone, 1990:11). Finally, unlike normal MRP II applications, the interface between maintenance and its supplier (DS) and customer (MM) will be automated within the logistics management systems modernization umbrella (Demmy and Giambrone, 1990:11).

DMMIS Planning and Control. Major AFLC planning and control activities include material requirements projections; procurement of supplies, repair parts, and equipment; and warehousing, distribution,

and repair support (Demmy and Giambrone, 1990:8). While differences do exist between commercial MRP II applications and AFLC maintenance operations, the required planning and control functions are basically the same.

Front End. As with standard MRP II, production planning is the determination by management (typically MM) of the overall level of effort (production) required to satisfy demands. Also like traditional MRP II, demand management involves determining the sources and magnitude of demands (both programmed and nonprogrammed) for repair resources. The programmed workload, called programmed depot maintenance (PDM), can be forecasted fairly accurately; however, the nonprogrammed workloads are difficult to predict. In line with AFLC's readiness goal, materials must be available for both kinds of workloads, so realistic nonprogrammed workload estimates must be determined to ensure that adequate inventory is available (Demmy and Giambrone, 1990:10). Under DMMIS, however, demand planning and production management will differ since maintenance works with only one customer, the Directorate of Materiel Management (MM). MM is responsible for preparing reparable items forecasts, quarterly projections of world-wide repair requirements, covering a period of five years into the future. These forecasts include estimates of reparable availability by item, quantity, and time period based on estimated failure rates, historical failure rates, and intended system use (AFLC, 1990a:15). Once the estimates are complete, the item manager then negotiates with MA for the repair support. AFLC planning and control systems will also provide the capability for performing rough cut capacity planning (RCCP). DMMIS, using workload histories and resource availability information, will

incorporate standard MRP II simulation logic to test the impact of new or modified workloads (Demmy and Giambrone, 1990:10).

As with traditional MRP II, the DMMIS master production schedule is a statement of what is expected to be manufactured or repaired. Functions of the MPS will include serving as an input to MRP and CRP calculations, keeping priorities valid, facilitating order promising and providing the ability to verify the accuracy of higher management's RCCP (AFLC, 1990a:22). Prior to release of the MPS, schedulers and planners determine detailed schedules for new components (material planning) and check (roughly) whether the capacity of each work center is adequate to meet the detailed material plans (AFLC, 1990a:20). The MPS in the AFLC repair environment accounts for workloads negotiated with MM (previously planned and scheduled), nonprogrammed workloads (not previously scheduled), workload routings to other divisions, and emergency/priority workloads (MICAPs). Just like standard MRP II, the final result is a master schedule specifying the number of reparables required for a given time period (AFLC, 1990a:26).

Engine. Once the workloads have been negotiated and entered into the master production schedule, the MRP and CRP programs are run (Demmy and Giambrone, 1990:10). Inputs to the DMMIS MRP program are the same as for any standard MRP II system: inventory data, bills of material, and the MPS; however, a significant difference is the inability to use modular BOMs in the maintenance environment. Due to the high level of uncertainty regarding the necessary processing to be performed on an given reparable asset, DMMIS requires the use of planning BOM's. AFLC BOM's use material usage rates to determine the quantity of planned units that can be expected to be replaced during a

given process (AFLC, 1990a:38). The output of MRP is a detailed schedule for MA workloads and DS requests (AFLC, 1990a:26).

Using the material plan as a basis, the DMMIS CRP module will use the BOM, the master routing file, and work center information to calculate how many machine and labor hours are required per week (AFLC, 1990a:56). The master routing file is a list of all possible repair operations (necessary to return it to serviceable status) that could be performed on a reparable item (AFLC, 1990a:52). During E&I for each asset, a detailed routing file is determined that identifies specific routing to be used for that particular item (AFLC, 1990a:53).

Back End. Shop-floor control encompasses execution of the repair/manufacture plan by 1) planning layout and flow, 2) controlling capacity and priorities, and 3) meeting quality, delivery, and production performance targets (AFLC, 1990a:62). Many of the DMMIS shop-floor control functions are identical to those in standard manufacturing. The system will support the release and tracking of repair and assembly orders, will provide up-to-date shop schedules and detailed short-range capacity projections, and it will support cycle counting and inventory analysis (Demmy and Giambrone, 1990:10-11). The typical DMMIS shop-floor control procedure is as follows: the order is planned and a due date is identified; the order is prioritized with respect to other planned orders and capacity; the order is released; the order is dispatched (a dispatch list is released daily); progress of the order is reported back to the planning stages; the schedule is adjusted as necessary; and the order is closed (AFLC, 1990a:67).

Problems/Criticisms of MRP-Based Approaches

The success of firms implementing MRP II is commonly classified into A, B, C, and D ratings. Those companies that use MRP II to its fullest and have achieved superior scheduling capabilities are termed "Class A" users (Teresko, 1986:40). Companies which use MRP II with progressively lesser degrees of success are termed "Class B, Class C, and Class D" users respectively (Teresko, 1986:40). The overwhelming majority of firms attempting to implement MRP II fail to reach Class A status (Cox, 1981:386; Clark, 1982:15). The literature indicates that less than 10 percent of users attain Class A status (Anderson, *et al.*, 1982:59).

There is a rising tide of disappointment with MRP-based methods and growing concern that MRP may not be an appropriate solution for manufacturing planning and control (Kanet, 1988:57). MRP consultants cite many reasons for the failure of MRP systems, including inaccurate computer records, unrealistic master production schedules, lack of top management involvement, and insufficient training; however, even when all these problems have been corrected, many MRP systems still fail to provide the advertised benefits (Kanet, 1988:58). In fact, continuing late orders, reliance on expediting, excess work-in-process, wandering bottlenecks, and invalid schedules still plague many firms that have "successfully" implemented MRP-based systems (Umble and Srikanth, 1990:9). The original intent behind MRP was to expedite materials when their lack would delay the master production schedule and then de-expedite materials when they were ahead of schedule; however, what typically happens is only the former, resulting in unneeded materials arriving at work stations too early (Chase and Aquilano, 1989:627).

Some critics charge that MRP simply computerized traditional manufacturing practices without actually improving the product flow through the plant (Umble and Srikanth, 1990:10).

Lead Times. One major problem area frequently noted by MRP II experts pertains to lead times. As discussed previously, all MRP systems use push-type scheduling methodologies. Using predetermined lead times for each process, the MRP module "backs off" from the order due-date (backward scheduling) to decide when each stage of production will require more materials (Fox, 1984:60). Critics claim that predetermining lead times (before job sequencing) does not permit consideration of flow time variation attributable job sequencing (Kanet, 1988:59). Goldratt contends that MRP does nct truly schedule backwards. Instead, it zig zags back and forth in time during the explosion process without simultaneously considering capacity limitations (Goldratt, 1990a:233). For this reason, he asserts that MRP will not produce reliable schedules (Goldratt, 1990a:233). In addition, MRP assumes that lead times are static (that they do not vary with quantities or differing conditions) when in reality they are dynamic (Chase and Aquilano, 1989:657). For these reasons, MRP becomes less dependable as the type of work moves toward R&D or maintenance efforts, since these types of work are characterized by longer and more uncertain lead times (Chase and Aquilano, 1989:643).

Feedback. Another problem relates to the provision of feedback. There is little disagreement about the importance of clear and timely communication to the success of MRP II; however, there is much controversy concerning whether MRP II actually provides a *formal* feedback mechanism (Kilmer, 1986:20; Kanet, 1988:59). Many experts

believe that MRP II systems lack a formal feedback channel and instead rely on informal, "off-line" feedback (Kanet, 1988:59). Lack of a formal feedback mechanism not only promotes the use of more safety buffers, but also it prohibits firms from taking advantage of strategic market opportunities (Kanet, 1988:59). These safety buffers significantly increase response time to the customer, further limiting a firm's market response capability. The absence of adequate, timely feedback significantly affects the ability of the system to respond to changes. Planning by MRP-based systems is performed sequentially, with planning done at one level and execution done at a lower level. Since proper feedback is not provided to the initial planning functions, management often discovers that plans are infeasible when it is too late to recover (Kanet, 1988:59).

Capacity Considerations. Another problem area relates to capacity requirements planning. First, few MRP programs can recognize and reschedule to solve the infinite capacity problem (Berger, 1987:240-243). Furthermore, the failure to run CRP and MRP modules together when changes are made to the MPS often results in "floating" bottlenecks that are difficult to trace (Berger, 1987:240-243). MRP does not consider the capacity problem until *after* the schedule is produced, introducing some questions regarding the realism of its schedules (Goldratt, 1990a:233). It is well-known by manufacturing personnel that schedules produced by MRP systems are not realistic, as evidenced by the term "infinite capacity" (Goldratt, 1990a:182). Even though newer MRP systems attempt to consider capacity limitations through a closed-loop process, in practice capacity is not usually considered, and the resulting schedules are not very realistic. The bottom-line: capacity

planning is still basically the responsibility of the individual designated as the master scheduler.

Constraints and Disturbances. Another recognized problem is the failure of MRP systems to account for constraints and disturbances in the production process which occurs after schedule generation. Goldratt suggests that any scheduling system must concentrate on the bottlenecks, thus reducing the amount of paperwork/complexity to a manageable level. In contrast to the TOC scheduling approach, MRP attempts to obtain information from all processing stations. MRP then attempts to control each individual operation, assuming that such control will lead to proper control of the plant as a whole. Unfortunately, this premise is not valid given the interactive nature of dependent operations (Umble and Srikanth, 1990:164). Furthermore, the task of compiling all the data required by MRP II is an enormous undertaking (Umble and Srikanth, 1990:164).

In addition, instead of attempting to produce schedules that are immunized from the effects of disturbances, MRP systems buffer almost every operation, leading to unnecessary inventory and exaggerated lead times. Goldratt contends that any viable scheduling system must account for the occurrence of disruptions in the production system. These disruptions, grouped somewhat unceremoniously under the name "Murphy," include such things as machine breakdowns or late shipments from suppliers. Goldratt contends that MRP cannot produce reliable schedules without properly accounting for Murphy.

Processing Speed. Schedule generation time under MRP takes at least hours, and often days (Goldratt, 1990a:165). One reason MRP is slow is that it is totally input/output (I/O) bound. Rather than

performing calculations, MRP prefers to store everything to disk and retrieve the information (Goldratt, 1990a:165). At the time MRP was developed, the I/O nature of its processing was a necessity--its designers did not have the tremendous amounts of on-line memory that are readily available today (Goldratt, 1990a:166). For a computer, recalculating figures is much faster (millionths versus hundredths of a second) than retrieving previously calculated figures. Unfortunately, many of today's commercial MRP II packages are still based on the technological limitations of the past--they are almost completely I/O bound: their actual computation time is only a small percentage of required scheduling time (Kanet, 1988:60). Correcting the disparity between calculation and retrieval speed can save as much as 1000 times the time required (Kanet, 1988:60).

Another reason MRP is slow is due to its awkward file structure. The most time-consuming part of scheduling is the explosion process. Today the product structure is normally split between a BOM and a routings file, causing a need to jump back and forth between the separate files during the explosion. The reason for the separation of BOM and routing files today is only inertia--it is a carryover of the technical limitations of the past (Goldratt, 1990a:170). When MRP was begun, the best available storage medium was magnetic tape. These tapes had to be processed sequentially, so splitting a portion of the data into a separate file resulted in faster processing times, since at that time it was quicker to switch between files than to reposition (rewind) on one large tape (Goldratt, 1990a:170).

The designers of MRP had three basic options: 1) include a complete description of the product everywhere it is needed in the

product structure, 2) have only one complete description and reference it in other necessary locations, or 3) separate the data into BOM and routing files (Goldratt, 1990a:170). If a complete description of the product was included everywhere, data integrity and maintenance would be a problem. Option two was not practical with tape processing, since the tape would have to be rewound every time another product required the data. Facing these choices, designers decided to create separate concepts for BOM and routings (Goldratt, 1990a:171). The major structural details of assembly are repeated everywhere, the BOM, and the vast majority of the detailed data is stored in one place, the routings.

The availability of direct access disks today has completely removed the technical limitations that caused the need for splitting the two files; however, BOM and routing remain separated (Goldratt, 1990a:173). The natural structure of the data should be one detailed description, with many references (Goldratt, 1990a:173). MRP designers also chose to split work in process and stores inventory into separate files, coding the WIP inventory according to the name of the next process, and the stores inventory according to the name of the last process (Goldratt, 1990a:175). The requirement to continually switch between the various files needlessly slows down MRP processing.

DMMIS Specific Concerns. All of the foregoing concerns are applicable to the implementation of any MRP-based system, including DMMIS. In addition to these general problems, some problems will likely be amplified within the DMMIS environment. Probably the most serious concern involves the higher uncertainty inherent in maintenance versus manufacturing activities. This high degree of uncertainty prohibits the use of standard, modular bills of material to forecast maintenance

workloads. In addition, the variation in processing and routing requirements for each reparable asset makes predetermination of lead times even more difficult. The result is that, unlike standard MRP II systems, MRP and CRP calculations in DMMIS will likely result in plans that only *estimate* material and capacity required to produce the planned workload (Demmy and Giambrone, 1990:10). For this reason, material and capacity plans must be continuously updated and monitored to avoid potential problems (Demmy and Giambrone, 1990:10). The necessity to continually rerun the MRP and CRP programs presents another issue for DMMIS. For the reasons discussed above, even with standard MRP systems, computer run time is significant. The long processing times required to run the MRP and CRP modules will likely be even more significant since these programs will need to be run "continuously."

Summary of Problems. As a result of these concerns, many plants that use MRP still tend to keep larger buffer inventories "just in case" there are disruptions in a shop or with a supplier (Cusumano, 1988:36). Some experts assert that the real problem lies with MRP itself--that MRP-based systems do not actually provide for production control, but only track things through the system without considering how to assign jobs based on developments within the system (i.e., congestion, machine downtime, etc.) (Chase and Aquilano, 1989:658). According to Sandman, a true production control system must provide "a plan to reach an objective, work assignments to meet the plan, and feedback to improve the quality of the plan" (Sandman, 1980:62-65). Critics such as Goldratt claim that MRP is "an excellent database, but it does not provide adequate scheduling" (Chase and Aquilano, 1989:659). He claims that despite 30 years of trying and the fact that producing reliable

schedules was a major objective, MRP is not a scheduler (Goldratt, 1990a:163). He posits that a major reason for the failure of MRP is that it is based on a faulty decision process, and that without the appropriate decision process, continuing efforts to make MRP a scheduling system have resulted only in extending the availability of data (Goldratt, 1990a:163).

To provide a true scheduling capability, there must be a simulation control system that provides a "daily work schedule sequenced job by job, work center by work center and hour by hour, and the capability to look ahead at future jobs" (Chase and Aquilano, 1989:658). Kanet proposes that instead of continually trying to "bandaid" the current MRP-based systems, new technologies need to be implemented that exploit the capability of computers and support decision-making instead of merely reporting on or accounting for it (Kanet, 1988:60).

JIT Manufacturing

Introduction. Thus far, this review has discussed the background, operation, and problems concerning MRP and MRP II systems. In addition, the DMMIS system operation has been examined. Next, the research will focus on just-in-time (JIT) manufacturing and the theory of constraints (TOC). Due to the similarity of JIT and TOC, this research briefly introduces the major principles of the JIT.

Discussion. Beginning with Toyota in the 1940's and 1950's, Japanese manufacturers made "critical changes to traditional U.S. manufacturing procedures that resulted in lower in-process inventory, higher inventory turnover rates, greater flexibility in equipment and labor, better quality, and higher overall productivity" (Cusumano, 1988:30). Toyota's new approach to manufacturing, titled JIT

manufacturing, made several departures from the U.S. approach. JIT stresses faster machine setup times, tighter synchronization between parts deliveries and assembly operations, lower inventory levels, and less specialization of workers and machines (enabling them to perform a variety of functions) (Cusumano, 1988:32). Although the Japanese made JIT famous, it is not a Japanese technology. Most of the principles of JIT originated in the United States (Hay, 1988:11). Furthermore, it is not the dominant manufacturing system in Japan, even though most Japanese businesses are now taking steps to incorporate it (Hay, 1988:10). The dramatic reduction in inventory levels realized by JIT manufacturers led to tremendous increases in throughput and quality for Japanese manufactured products. These improvements have prompted many U.S. companies to focus more attention on new manufacturing technologies.

JIT is a methodology for removing non-value-adding activities from manufacturing, distribution, and purchasing (Hay, 1988:1). There are three major components to the JIT philosophy of management: quality at the source, changing (improving) the production process to ensure a smooth flow of the product from each operation in the organization, and obtaining employee commitment and involvement. A major objective of JIT is the elimination of waste. The JIT definition of waste is anything that "does not add value to the product" or "anything other than the minimum amount of equipment, materials, parts, and working time absolutely necessary for production" (Hay, 1988:15). Unlike the traditional concept of waste, the JIT definition considers activities such as inspection and scheduling to be "waste" since they add no value to the product.

Another major tenet of JIT is "producing precisely the necessary units in the right quantities at the right time" (Chase and Aquilano, 1989:743). Producing one extra unit is as bad as producing one too few because excess WIP will incur additional expense to store and maintain (Chase and Aquilano, 1989:743). Unlike their American counterparts, the Japanese believe that permitting a worker to sit idle is not a crime, but having idle material is. The American approach has traditionally been just the opposite--keep people busy no matter how much excess inventory they produce (Fox, 1984:56). Under JIT, managers do not concern themselves with achieving rated equipment speeds--they produce only the amounts needed at the present (Chase and Aquilano, 1989:746). JIT does not allow for contingencies--managers intentionally drive inventory levels as low as possible. According to JIT, high inventory levels only hide problems, and lowering the inventory level enables managers to identify (and fix) problems (Chase and Aquilano, 1989:745).

JIT was the first manufacturing technology to employ a "pull" system for scheduling (Cusumano, 1988:34). One process, called the kanban system, uses kanbans (cards) as authorization to produce (or withdraw) another container of parts after a work station empties the original container (Bylinski, 1983:126). Under kanban, authority to produce comes from someone downstream in the production process, thus the system "pulls" material through the production process. Managers plan work according to a schedule, but only execute the plan (produce) in accordance with the kanban (a completely manual system). The basic idea is that when workers need more parts, they travel to the preceding work station and 1) insert a withdrawal kanban (card) authorizing them to remove the container, 2) remove the production kanban associated with

the full container, and 3) place the production kanban on a rack to authorize production of another container (Chase and Aquilano, 1989:747). The order of the production kanbans on the rack determines the priority of production, and the container size determines the quantity (Chase and Aquilano, 1989:747).

The Theory of Constraints

Scope. A new philosophy that is rapidly gaining popularity in today's manufacturing community is synchronous manufacturing, based on Goldratt's *theory of constraints*. The intent of this discussion is not to provide a complete review of TOC: Trigger provided a comprehensive discussion of the background and principles of the theory in his 1990 thesis. The purpose of this discussion is to review and highlight aspects of TOC that one must understand to comprehend the logic behind the *DISASTER*™ scheduling information system.

Introduction. The *DISASTER*™ scheduling system is based directly on the *theory of constraints*. The basic concepts behind *DISASTER*™ are not new: in the early 1980s Goldratt marketed a similar software system, called Optimized Production Technology (OPT), that used the same basic logic as *DISASTER*™ (Bylinski, 1983:121; Trigger, 1990:29-30). In addition, many of the principles of TOC are shared by other contemporary management philosophies such as JIT and Total Quality Management (TQM). Many experts lauded the performance of the original OPT system; however, it failed to secure a place in the market, due mainly to the absence of a concerted effort to disseminate the "thoughtware" necessary for its success (Trigger, 1990:27). Instead, OPT required users to invest up to $500,000 for the software with no understanding of its internal processing logic (Trigger, 1990:27).

As discussed by Goldratt in his book, *What is This Thing Called the Theory of Constraints and How Should it be Implemented*, the change to a new management philosophy such as TOC requires a significant cultural change by an organization's personnel. Despite the importance of human psychology in any major system change, most management science literature ignores the impact of change on an organization's personnel (Trigger, 1990:44). Goldratt, recognizing the shortcomings of his original OPT marketing effort, has put much effort into disseminating his theory *prior* to releasing *DISASTER™*. Considerable thought was put into the cultural changes that are required for *DISASTER™* to be a success. Today's American managers rely heavily upon standard cost accounting principles and logic. This "cost world" is deeply ingrained in American managers, yet the decision process used by the cost world is often inappropriate for a throughput-based information system such as *DISASTER™*. Goldratt advocates use of the "Socratic approach" to overcome resistance to change: if one gets the users to think of the necessary changes and make the right decisions themselves, they will then possess the emotion required to overcome any resistance to change (Goldratt, 1990b:10-16). Since *DISASTER™* is based on a significantly different management philosophy, Goldratt warns that the organization must undergo the requisite cultural change *before* implementation (Goldratt, 1990a:10). Use of an information system based on TOC is entirely dependent upon how well the users understand and accept the principles behind the philosophy (Goldratt, 1990a:88,105).

As noted above, while TOC does involve a significant change in culture, it is not an entirely new philosophy. Like JIT, TOC aims for small production lot sizes, and concentrates on minimizing inventory;

however, unlike JIT, TOC uses extensive computer control to develop schedules (first with Optimized Production Technology and today with *DISASTER*™) rather than a manual system like the Japanese kanban (Bylinski, 1983:126). In addition, TOC focuses more attention on ensuring that the organization performs in a manner consistent with its overall goal and scheduling the flow of material by identifying and properly managing bottleneck resources to synchronize the production flow.

TOC stresses that the "goal of any manufacturing organization is not to employ workers, keep machines busy, or provide a service, but rather to make money, and everything else is subordinate to that goal" (Edwards and Heard, 1984:45). While TOC can be applied to any type of organization, the majority of its efforts have been directed towards manufacturing. The theory can easily be applied to AFLC manufacturing operations, since a manufacturing manager's objective is to manage the manufacturing operation such that market demands are met at minimum cost (Umble and Srikanth, 1990:133). Even though industrially-funded organizations do not operate to make money, they do strive to minimize operating expense and overall losses while meeting the users' demands.

A key principle behind TOC is the idea that bottlenecks are what constrain manufacturing output (Powell, 1984:100). Goldratt believes that there are two types of resources in any plant: bottlenecks (or constraints) and non-bottlenecks (or non-constraints). Bottlenecks are resources in the manufacturing process that limit the amount of product that a factory can produce, and nonbottlenecks are all other resources owned by the plant (Bylinski, 1983:121). A principle rule of TOC states that the utilization of a nonbottleneck resource cannot be controlled by

its own capacity, but must instead be determined by other constraints in the system (Fox, 1984:56-57). In other words, under normal circumstances, nonbottleneck resources should not be operated at a level higher than needed to support the capacity of bottleneck resources. In contrast, bottlenecks must be identified and operated at maximum capacity because they govern the throughput of the entire system (Fox, 1984:57).

Traditional manufacturing philosophy has ignored the effects of bottlenecks in production operations, instead driving all workers and machines to operate at maximum efficiency (Bylinski, 1983:121). In contrast, by synchronizing production, TOC strives to balance flow, not capacity (Fox, 1984:56). The recognition of the importance of bottleneck resources ensures that TOC-managed plants do not operate nonbottleneck resources at maximum capacity when their work only creates excess inventory (Powell, 1984:100). In addition, waiting between jobs (queue time) accounts for as much as 99 percent of the time an item spends in the production process; therefore, TOC (as does JIT) stresses that items should arrive at each work station as close as possible to the time actually needed for production (Bylinski, 1983:124).

Perhaps the greatest impact of TOC has been in the scheduling of operations on the shop floor for an increasing number of U.S. firms including Ford, General Electric, General Motors, Westinghouse, and RCA (Melton, 1986:13). As discussed previously, MRP is generally regarded as using backward scheduling (even though it does not move consistently backward), basing everything on a master production schedule and predetermined lead times for each production activity. It is very difficult to schedule production so that flow through the plant is

smooth (balanced). The result is more local decisions that deviate from the master schedule (such as unplanned expediting) and quickly render the schedule invalid (Chase and Aquilano, 1989:802). Even under JIT, managers tend to adjust production rates only *after* buildups occur (Powell, 1984:100). In contrast, *DISASTER*™ schedules consistently backward in time while simultaneously considering capacity limitations. This practice ensures that all loads on resources are within capacity. Furthermore, by performing computer simulations that consider all constraints simultaneously, *DISASTER*™ determines in advance how a change at any stage of the process will affect production. For these reasons, many people believe that TOC can produce an even smoother manufacturing flow than JIT, resulting in lower work-in-process inventory levels, higher throughput, and reduced operating expense (Chase and Aquilano, 1989:802).

New Performance Measures. The TOC philosophy "lays waste to the notions of efficiency that have been used to manage manufacturing systems for the past 75 years" (Powell, 1984:100). Similar to JIT philosophy, TOC stresses that considerations such as cost per unit and efficient (full) utilization of resources (beyond the level actually needed) leads to the buildup of excess work-in-process (WIP) inventories (Edwards and Heard, 1984:46). People often have distorted views concerning the question "what is the goal?" According to TOC, the goal of an organization is not to produce parts or repair components, but rather to make money, and the important measure is the *rate* at which the system generates money, not the absolute amount (Goldratt, 1990a:17). Goldratt believes that one of the foundations of a company is its ability to judge the effect of decisions on the bottom line, so he

stresses that all measures must have a dollar sign in front of them (otherwise one would be comparing apples and oranges) (Goldratt, 1990a:55).

Goldratt, as well as a growing number of respected management accountants, believes that manufacturing operations managed under today's new philosophies need new operational and performance measures (Edwards and Heard, 1984:44). According to Goldratt, if a TOC organization evaluates its personnel using traditional measures such as efficiencies, these measures will prompt behavior in conflict with the goal: "tell me how you will measure me and I will tell you how I will behave. . .if you measure me in an illogical way, I will behave illogically" (Goldratt, 1990a:26). TOC places much importance on the identification of the goal because any decision or action should only be judged by its effect on the system's goal--in order to improve performance relative to the goal, one must know precisely what that goal is (Trigger, 1990:30). Performance measurements provide the "bridge" between actions and their impact on the goal (Trigger, 1990:31). For this reason, TOC (unlike other new management philosophies such as JIT and TQM) proposes three measures that should be the primary considerations of any manufacturing management accounting system: throughput, inventory and operating expense.

Throughput is the rate at which the system generates money through sales (Goldratt and Fox, 1986:28). The important aspects of this definition are 1) that it is the *rate* at which dollars are generated (not the number of widgets, service completions, etc.), and 2) that throughput is not increased unless products are actually *sold* (Goldratt and Fox, 1986:28). Throughput is not analogous with sales. It is equal

to the selling price minus the amounts paid to vendors for items that went into the product sold (Goldratt, 1990a:20). This definition requires that one determine the point in time when the sale occurs (Goldratt, 1990a:20).

The second measurement, inventory, is defined as "all the money the system invests in purchasing things the system intends to sell" (Goldratt and Fox, 1986:28). Inventory is basically the same as the conventional business definition, but unlike convention, this definition includes buildings and equipment (Goldratt, 1990a:23). Due to the negative effects of WIP, Goldratt intentionally avoids the term assets to refer to inventory. According to the TOC definition, inventory is valued at only the price paid to vendors for the material and parts purchased that went into the product--value added is not recognized (Goldratt, 1990a:23). TOC is not concerned with value added to the product, but only with value added to the *company*, and value is added to the company only when the product is sold (when throughput is realized) (Goldratt, 1990a:24). Clearly, inventory should be considered a liability, yet under conventional manufacturing systems plant managers are still evaluated with inventory under the asset column--in a normal American company, if a manager makes a significant reduction in inventory, it appears that he has lost a portion of the net worth of the company. Furthermore, if a manager underutilizes the work force or equipment, he or she will again be evaluated negatively. The inappropriate evaluation of managers (as in the above example) hinders inventory reduction efforts in American companies, again highlighting the need for a total company cultural change (Goldratt, 1990a:25).

Operating expense (OE) is "all the money the system spends in turning inventory into throughput" (Goldratt and Fox, 1986:28). In contrast to conventional definitions of inventory and operating expense, dollars are *invested* in inventory and *spent* on operating expense. When a purchase of materials is made (for instance, oil) then it is initially classified as inventory, then reclassified as OE as it is used. Likewise, a purchase of material used in the product is first termed inventory, then as some of it is scrapped, it is reclassified to OE (Goldratt, 1990a:29).

Instead of evaluating decisions based on any *one* measure alone, one must consider the relationships between all three measures (Goldratt, 1990a:32). These measures are not new in themselves; however, their order of importance is different in TOC versus traditional cost-world manufacturing (Trigger, 1990:32). The natural tendency by cost-world managers is to place emphasis on OE and to attempt to improve by reducing it. Cost world thinking considers only the impact of inventory carrying cost and depreciation of capital and equipment--the effect inventory has on OE & the bottom line (Goldratt, 1990a:50). The result of only considering these indirect impacts is that inventory is placed below OE on the cost-world scale of importance (Goldratt, 1990a:50). The cost-world scale of importance for the three measures is operating expense then throughput, with inventory a remote third (Goldratt, 1990a:49).

JIT, TQM, and TOC all have the objective "continual process improvement" in common. If one wants a process of continual improvement, then TOC advocates putting throughput at the top of the scale of importance: inventory and OE are both limited by zero, yet

throughput is unlimited (Goldratt, 1990a:49). According to TOC, the new scale of importance is throughput, followed by inventory, then operating expense (Goldratt, 1990a:51). TOC places the importance of inventory over operating expense, as does JIT and TQM, because it recognizes that inventory accumulations have another direct impact: excessive inventory affects the time-related performance of the company, and therefore it directly impact a company's ability to compete (Goldratt and Fox, 1986:32). This new scale of importance is completely different than the cost world scale, and it changes many of the resulting decisions (i.e., make or buy, pursuit of particular products, etc.).

One reason for the cost-world's focus on OE is the unverbalized assumption that systems are comprised of independent variables. This premise leads to the belief that minimizing the operating expense of each of these variables will minimize the operating cost of the system as a whole. Another reason TOC proposes to explain the preoccupation with OE is that costs are readily identifiable--managers are more accustomed to identifying and accounting for costs using well-established cost accounting procedures (Trigger, 1990:65). This focus on OE often results in an unmanageable system, for there are numerous "outlets" for operating expense. Even when the Pareto principle is employed, it is still very difficult to determine where a manager should focus his or her efforts (Goldratt, 1990a:52).

Instead of viewing our systems as systems of independent variables, the throughput world views them as systems of dependent variables (Chase and Aquilano, 1989:798). When throughput is the dominant measure, one can see that many tasks have to work synchronously to produce the product. In a sense, the throughput world views these

tasks as chains, and the weakest link in the chain determines the throughput potential of the entire chain. This new scale of importance is a foundation of TOC, for it permits TOC managers to focus attention on these "weak links." In effect, this new scale of importance leads to a new Pareto Principle: instead of 20-80, under TOC the ratio is closer to .1-99.99 (Goldratt, 1990a:53).

The Cost-World Versus the Throughput World.

Origin of Cost Accounting. Cost accounting was developed to enable managers to determine the effect of local decisions on the company as a whole by breaking down expense categories product-by-product. The principal idea behind cost accounting is the approximation of the direct labor contribution by product and use of this estimate to evaluate local decisions. Any additional expenses that cannot be directly applied to a product are labeled as "overhead" and allocated according to the contribution of direct labor to each product. Splitting OE product-by-product enables managers to evaluate specific products without looking at other products. Instead of trying to split costs by product, TOC posits that net profit can only be calculated for the company as a whole, not for individual products (Goldratt, 1990a:42). Apportioning direct and indirect costs to each product was appropriate when cost accounting was developed, at the turn of the century; however, the declining percentage of direct labor to overhead has invalidated the original foundation of cost accounting (Drucker, 1990:97).

According to Drucker, the limitations of cost accounting were evident as early as the end of World War II (Drucker, 1990:97). Today, the situation is even worse: direct labor accounts for only about 8-12

percent of total cost, yet the remaining costs are still allocated based on "purely arbitrary and misleading ratios such as a product's direct labor costs" (Drucker, 1990:97). Another limitation of cost accounting is its failure to recognize the costs of nonproducing, whether from machine downtime or quality defects (Drucker, 1990:97). Finally, Drucker stresses that manufacturing cost accounting incorrectly treats the factory as an isolated entity, and this fact prohibits the methodology from justifying product or process improvement (Drucker, 1990:97).

The Impact of Cost-World Thinking. Goldratt often uses *Gedunken* (the German word for "thinking") experiments to illustrate many of his points. One of his common Gedunken experiments portrays the difference between the cost world and the throughput world and the importance of recognizing the proper scale of importance for the operational measures. Consider Figure 4, adapted from *The Haystack Syndrome*, describing the production of two products, P and Q (Goldratt, 1990a:67). The letters A through D represent each of the plant's resources, which in this case are workers. Each worker is available 5 days per week, 8 hours per day, and 60 minutes per hour. The market potential per week is 50 units for product Q and 100 units for product P. Total plant operating expense is $6,000 per week.

Using this information, Goldratt asks manufacturing managers to compute the net profit per week for this plant. The most common answer to the problem is derived as follows:

Throughput for P is $90 - $45 (total raw material cost) = $45.
Throughput for part Q is $100 - $40 = $60
50 units of Q (market demand) X $60/unit = $3,000
100 units of P X $45/unit = $4,500
Total net profit = $7,500-$6,000 OE = $1,500

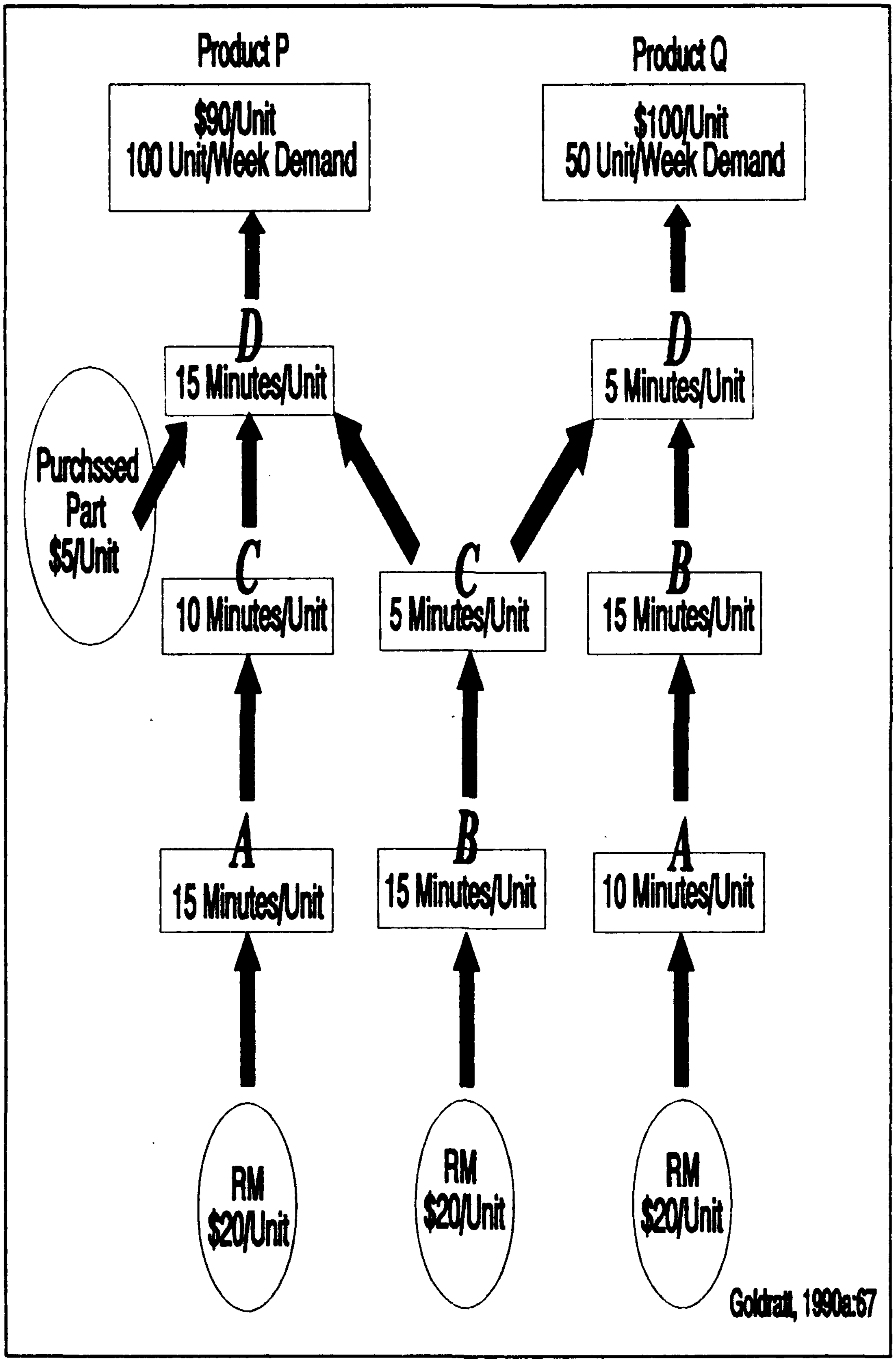

Figure 4. Product P and Product Q Gedunken Experiment

Many cost-world manufacturing managers fail to recognize the existence of any constraints in the process, and they arrive at a net profit figure of $1,500.

Even when managers recognize the existence of internal constraints (i.e., processing time available from each of the workers), they still fail to fully consider the importance of processing time on the constraint. Notice that production by resource (worker) B requires 3,000 minutes to produce all of the market potential for both products (100 units of P times 15 minutes/product plus 50 units of Q times 30 minutes/product); however, only 2400 minutes are available for processing. Clearly, a decision must be made as to which product to produce first, and which to produce with the remaining capacity. Every cost-world indicator (sales price, raw material cost, labor content, product margin, etc.) would lead the manager to produce all the Q's, then concentrate on the P's. Nevertheless, it takes 30 minutes of effort (on the constraint) to produce a unit of product P and 15 minutes of effort to produce a unit of product Q. Most managers will then choose to produce 50 units of product Q and use any remaining capacity to produce product P. Fifty units of Q X 30 minutes/unit requires 1500 constraint processing minutes, leaving 900 minutes available for producing units of product P. Thus, 50 units of product Q and 900 divided by 15, or 60 units of product P, can be produced. Net profit calculation is now:

50 Q's X $60/unit =	$3,000
60 P's X $45/Unit =	$2,700
Total Throughput =	$5,700
Less OE	$6,000
Net Profit	($300)

This last method is still not entirely in line with the throughput world, since it still uses *product* profit (Goldratt, 1990a:76). The correct analysis must consider the amount of money that will be earned per minute of processing by the constraint. Using this technique, one generates $3 per minute ($45/15 minutes) for product P, and $2 per minute ($60/30 minutes) for product Q. Using this analysis, product P is definitely more profitable than product Q, thus managers should maximize the amount of P they can offer the market (Goldratt, 1990a:77). Net profit is then calculated as follows: 100 units of P times $45 = $4,500, and these units will require 1500 minutes of the constraint; using the remaining 900 minutes, 30 units of Q can be produced, generating an additional 30 X $60/unit = $1,800 in throughput. Net profit is then equal to ($4,500 + $1,800) - $6,000 = $300.

Most of today's managers are managing according to the cost world. They spend a majority of their time "putting out fires," believing that nearly everything is important. Goldratt asserts that top managers spend 80 percent of their time "putting out fires" (Goldratt, 1990b:63). According to TOC, managers need to undergo a fundamental change from cost world thinking to the throughput world--they need to quit spending their attention on too many "seemingly equally important problems" (Goldratt, 1990a:54). In addition, managers should discontinue use of financial measures that conflict with the overall company goal. For example, consider decisions on whether to drop a product. In cost-world thinking, considering product profit (local optimum), top management may conclude that the "unprofitable" product should be dropped; however, overhead is seldom reduced (people are not laid off). The high percentage of fixed to variable cost prevalent in today's manufacturing

organizations only aggravates the problem (machines cannot be laid off). If the global throughput minus operating expense formula is used, as illustrated by the above gedunken experiment, this problem can easily be overcome (Goldratt, 1990a:45).

TOC Versus JIT and TQM. Like JIT and TQM, TOC is a new overall management philosophy, and as such it requires a commensurate cultural change. JIT and TQM have helped in forcing management to switch to the new scale of importance, but they have not prompted the necessary change in management philosophy (Goldratt, 1990a:54). TQM places customer service and product quality first (Goldratt, 1990a:54). JIT has changed the perception that inventory is an asset (Hay, 1988:31). In addition, JIT recognizes the importance of shrinking production lead time, reducing batches and setup times, and improving preventive maintenance. While both JIT and TQM have realized the importance of throughput, they have both failed to fully recognize the nature of the throughput world, especially with respect to the dependent nature of systems and the need to focus on the system's constraints. It is not equally important to decrease all setups, only those on the constraint. The idea that all setups are equally important is a carry-over from the cost world (Goldratt, 1990a:54).

Another failing of JIT and TQM is that they have not attempted to establish financial measures that can be used to evaluate decisions (Goldratt, 1990a:55). JIT ignores the issue and TQM encourages nonfinancial measures such as "quality is job one" (Goldratt, 1990a:55). The result of TQM's and JIT's failure to fully recognize the dependent nature of manufacturing systems and their tendency to focus on local versus global optima is that they are incapable of identifying areas for

process improvement. Both philosophies advocate that continual process improvement is necessary, yet fail to provide a systematic method for determining what should be improved or changed.

The TOC Decision Process. In contrast to JIT and TQM, TOC recognizes the need for a systematic method of determining *what* processes need to be changed. A major problem with the cost world is its tendency to focus on everything, or, stated otherwise, its failure to focus on anything. The throughput world has a much narrower focus. It concentrates on just the constraints, since they determine the overall performance of the company (Goldratt, 1990a:58). The only time a company faces a problem is when they lack something--when they face a constraint (Goldratt, 1990a:55). A key to TOC is the focus it provides on an organization's constraints, allowing managers to decide where their attention should be focused (Trigger, 1990:33). TOC places much emphasis on the development of a good decision process that will permit a company to focus on their constraints and identify areas for process improvement.

Goldratt has developed a five-step focusing procedure. The first step in the TOC focusing process is to identify something that is definitely a constraint (Goldratt, 1990b:5). It is not important at this stage to prioritize the constraints. The focusing steps are an iterative process, identifying additional constraints with each repetition until all constraints have been identified. Every system must have at least one constraint, yet there will only be a limited number of constraints, so the iterative nature of the process is not a problem (Goldratt, 1990a:59). The second step in the focusing process is to exploit the constraint: to determine how to "squeeze" the maximum

possible out of the constraint (Goldratt, 1990b:5). The third step is to subordinate everything else to the above decision--to subordinate all non-constraint resources to support the constraints. The system needs to ensure that the non-constraints do not supply more than the constraint can handle (Goldratt, 1990a:61). The fourth step is to elevate the system's constraint(s)--to open them up and allow them to supply more (Goldratt, 1990b:5). Often people jump from the first to the fourth step; however, this practice should be avoided until after the exploit and subordination steps have been completed (Goldratt, 1990a:61). Eventually, as the constraint is opened, it will cease to be the constraint--something else will become the weak link (Goldratt, 1990a:62). This leads to the fifth step: if in the previous steps a constraint is broken, go back to step one and start again (Goldratt, 1990b:6). Goldratt warns that during this process, rules (policy constraints) are often imposed to facilitate exploitation of the constraint. Once this constraint is broken, these policy constraints are no longer needed: however, due to "inertia," they are often left in place (Goldratt, 1990a:62). To guard against this inertia, Goldratt splits the final step into a second stage: do not let inertia become the system's constraint (Goldratt, 1990a:62).

Process Improvement. The need for ongoing improvement is a central tenet of TOC (Trigger, 1990:40). Goldratt and Fox emphasize the importance of such improvement in *The Race*:

> Those unable to *continually improve* are falling behind, since success in this environment requires more than a one-time investment. . . Clearly, something far greater than a few sporadic improvements is now needed. Indeed, the only way to secure and improve one's competitive position today is by instituting a process of ongoing improvement. (Goldratt and Fox, 1986:144)

The obvious place to look for improvement is with the constraints, not the non-constraints. Improvements in the nonbottlenecks will only produce more idle time, but improvements on the constraints yield more throughput for the entire company (Chase and Aquilano, 1989:806). The data required to investigate such improvements is simply the identification of the constraints and how many dollars per minute can be squeezed out of the constraint (Goldratt, 1990a:91).

The third focusing step, subordination, requires a drastic change in management philosophy, especially with respect to performance measures (Goldratt, 1990a:87). If properly performed, subordination requires that some resources (the non-constraints) will work much less than their capacity. This underutilization directly conflicts with the current management philosophy of maximizing efficiency. The fact that resources are idle should lead directly to the idea of process improvement: workers then have more time to identify and implement improvements. Furthermore, Goldratt contends that no system can survive with too many interacting constraints--there can only be a few constraints in any viable system (Trigger, 1990:35). This fact limits the number of areas on which TOC managers must focus, reducing the Pareto Principle under TOC to a ratio closer to .1 to 99.99. Use of the TOC focusing steps enables managers to concentrate on the important issues and problems: the constraints (Trigger, 1990:35).

Drum-Buffer-Rope. Unfortunately, most manufacturing plants today equate minimizing total cost with minimizing the cost of each individual product. Attempts to minimize the costs of products by reducing setups and increasing batch sizes conflict with the need for a smooth, fast material flow (i.e., smaller batches and more setups) (Umble and

Srikanth, 1990:133). In reality, total manufacturing cost cannot be minimized by minimizing individual product costs due to the complex interactions present in manufacturing operations. To deal with these new interactions, a new, manageable logistical control system is required (Umble and Srikanth, 1990:134). Drum-buffer-rope (DBR) provides the general guidelines for producing shop floor schedules using TOC concepts. There are two major issues that must be addressed to achieve synchronous manufacturing flow: the ability of the plant to execute the planned product flow over a given horizon and the impact of deviations on the planned product flow. Drum-buffer-rope considers both of these issues. (Umble and Srikanth, 1990:136). The basic strategy of DBR is as follows:

1. Develop an MPS based on the constraints of the system (drum).

2. Protect the throughput of the system from deviations with the use of time buffers at critical locations (buffer).

3. Tie the release of materials for production at the non-constraint resources to the rate at which they can be processed by the drum (rope).

The essence of DBR is that the production rate of the constraints serves as the "drum," signaling when material should be released to the floor (Trigger, 1990:69). To account for deviations that will inevitably occur in the manufacturing system, buffers are used to protect the throughput of the plant (Umble and Srikanth, 1990:137). The "rope" is then tied from the constraint to the first operations in the manufacturing process (the gating operations) to communicate the rate at which the system should produce (Goldratt and Fox, 1986:96-102).

According to Schragenheim and Ronen (1989a:12-13), DBR can be described as follows using the first three steps of the focusing procedure:

1. Since the performance of the firm, and the schedule, is controlled by the constraint, the first step is to identify it by calculating the cumulative demand placed on each resource relative to its capacity.

2. Exploitation involves taking all measures necessary to ensure the constraint is never idle (by using a constraint buffer) and ensuring that the constraint is working on the right product mix. In addition, other buffers may be required to ensure that items already processed on the constraint are not delayed (assembly or finished goods buffers).

3. Subordination directs the efforts of all other work centers towards accomplishing the schedule that was developed based on exploitation of the constraint.

The Drum. Every system needs a control point, and if the system contains a bottleneck, then the bottleneck is the logical point at which to control the system. The bottleneck is used as the control point because it will keep non-constraint resources from overproducing and building up excess WIP in front of the constraint (Chase and Aquilano, 1989:808). Since constraints control the firm's performance, in a sense they replace product cost as the primary tools of management (Goldratt, 1990a:55). Since the bottleneck schedule specifies the production rate for the entire system, TOC refers to it as the "drum" (Chase and Aquilano, 1989:808). The drum includes a detailed schedule of the products and quantities to be processed on the constraint (Trigger, 1990:72).

The critical constraints in a manufacturing system are market demand, capacity, and material limitations (Umble and Srikanth, 1990:137). It is critically important to focus on the constraints when authorizing the drum. Otherwise, the required capacity of the constraint may be greater than available capacity and planned product flow will be jeopardized (Umble and Srikanth, 1990:147). The first step of DBR is to derive a basic production plan that accounts for these constraints (Umble and Srikanth, 1990:137). If the market is the constraint, then the plan is simply the due dates of the products; however, if a resource is loaded to the point that some due-dates cannot be met (i.e., a resource is the constraint and some products may have to be given priority over others), then the plan is determined by giving priority to the products according to profit margin per unit of processing time on the constraint (Trigger, 1990:74-5). Once this plan has been developed, then a detailed schedule for the constraint is then is produced. The process used to derive this schedule is called authorizing the drum, and the result, the master production schedule (MPS), provides the sequence of product type and quantity to be processed on each constraint (Umble and Srikanth, 1990:137).

Buffer Management.

Statistical Fluctuations, Dependent Events, and Murphy. Statistical fluctuations cause variation in processing time about some mean. Dependent events are operations whose performance is contingent upon the completion of another event. When statistical fluctuations occur in a system of dependent processes without any inventory between work stations, there is no opportunity to achieve average output. When one process takes longer, there is no way for

subsequent processes to make up the time (Chase and Aquilano, 1989:798). Manufacturing processes are normally characterized by a wide distribution of output times from work stations. This characteristic causes downstream stations to have idle time when upstream stations take longer to process, (Chase and Aquilano, 1989:797). The effect of these two phenomena, statistical fluctuations and dependent events, is cumulative, and the interactive nature of dependent events causes disruptions to quickly spread throughout the manufacturing plant (Chase and Aquilano, 1989:797).

The significance of statistical fluctuations and dependent events is magnified by the occurrence of random events that affect the manufacturing process. TOC refers to such disturbances as "Murphy." A major concern is that Murphy cannot be predicted with any accuracy and its effects can never be entirely eliminated (Umble and Srikanth, 1990:52). TOC recognizes two types of Murphy: pure Murphy and non-instant availability (Goldratt, 1990a:127). Pure Murphy occurs when a machine breaks down, or a worker fails to show up for work. Non-instant availability occurs when a part arrives for processing, but the resource is not available--the part has to wait in queue (Goldratt, 1990a:127). Goldratt asserts that the majority of the time a product spends in the manufacturing system is spent in queue and can be attributed to noninstant availability.

Protective Capacity vs Protective Inventory. Since any system must have one "weakest link," other resources must by definition have additional capacity (Goldratt, 1990a:112). Managers have traditionally considered any capacity above the amount strictly required for production as excess capacity. In traditional cost-driven

systems, managers often focus on trimming this "excess" capacity as a means of reducing cost; however, elimination of this capacity also eliminates the system's capability to catch up when disturbances occur. The end result is that either the system will fall farther and farther behind the production plan, or managers will have to spend more money to produce excess capacity (i.e., through overtime or subcontracting) (Umble and Srikanth, 1990:63). Traditionally, manufacturing companies have attempted to balance capacity across all work stations; however, when a system is characterized by statistical fluctuations and dependent events (as most manufacturing systems are), the balancing of capacities is inappropriate (Chase and Aquilano, 1989:797). When statistical fluctuations exist (i.e., due to Murphy) among dependent resources, then all non-constraints need to have the capability to "catch-up" after disturbances occur. Additional capacity enables non-constraint resources to rebuild their buffers after Murphy hits (Goldratt, 1990a:113). If a non-constraint has no additional capacity, then each time Murphy hits, the constraint's protective inventory will be reduced, and the resource will not be able to work fast enough to replenish it. Eventually, the constraint's operations will be interrupted (Goldratt, 1990a:113). Based on these observations, it becomes apparent that any non-constraint resource with no additional capacity requires an infinite buffer.

TOC advocates balancing product flow rather than capacities (Chase and Aquilano, 1989:797). The theory distinguishes between three, as opposed to the traditional two, categories of capacity: productive capacity is the amount of capacity needed for actual production to meet demand; protective capacity is the amount needed to shield against

Murphy; and the remaining capacity is considered excess capacity to be considered as potential for additional throughput (Goldratt, 1990a:114). Since a constraint cannot have any protective capacity itself, it must be protected by a combination of inventory placed just in front of it and protective capacity of the non-constraint resources (Goldratt, 1990a:114). A tradeoff exists between the level of protective inventory and the level of protective capacity: reduce one, and the other must be increased (Goldratt, 1990a:114). This new concept of protective capacity directly impacts decisions such as whether or not to build items in-house. If the new product must be processed on any non-constraint without sufficient excess capacity, production of product can only be accomplished by using some of the resource's protective capacity. This reduction in protective capacity requires a counterbalancing increase in protective inventory (for any item that crosses the constraint in question) if the system is to keep the same level of protection (Goldratt, 1990a:114).

Buffers. Disruptions due to Murphy are not all equally important. A disruption at a non-constraint may disrupt the timing of the flow, but it will not impact the production of the product. In contrast, disruptions on a bottleneck resource directly impact throughput (Umble and Srikanth, 1990:139). To smooth the effect of disturbances, TOC requires that buffers be used to protect the constraint; however, only the minimum amount of material necessary for uninterrupted operation of the constraint is permitted to enter the system (Trigger, 1990:38). The few places where the effects of Murphy, as well as other statistical fluctuations, tend to accumulate are the inventory locations just in front of the constraint(s)--this is the

logical basis for what TOC calls buffer management (Goldratt, 1990a:119).

The occurrence of Murphy is not in question--disturbances will occur, so what needs to be determined is when, where, and how long they will last (Goldratt, 1990a:124). Furthermore, in most environments, disturbances are the overwhelming factor in determining a task's lead time--actual process time is negligible (Goldratt, 1990a:127). Any reliable scheduling system must account for Murphy and continually strive to reduce its impact (Umble and Srikanth, 1990:136). The traditional (MRP) approach to dealing with Murphy has been to accept its existence and to use inventory to buffer nearly every operation. TQM also recognizes the need to fight Murphy and asserts that one should not accept Murphy, but rather should try to eliminate it (Goldratt, 1990a:133). TOC's approach is more moderate than TQM's. It recognizes that Murphy cannot be eliminated, but rather should be minimized (Goldratt, 1990a:133). Rather than buffering every operation, TOC is very selective in its use of buffers. Recognizing that excessive buffers create excessive inventory and lead-time, TOC strictly controls the use of buffers (Goldratt, 1990a:134).

In contrast to the more conventional method of using *inventory* buffers, TOC uses *time* buffers to protect against unknown disturbances. In simple terms, time buffers represent the amount of time needed to protect the system against disruptions to the manufacturing flow. Time buffers increase the planned lead time from the absolute minimum necessary to produce the product to a time sufficient to shelter throughput from any disruptions (Umble and Srikanth, 1990:137). The action necessary to produce time buffers is prerelease of work, relative

to the date at which the corresponding constraint's consumption is scheduled. TOC uses time as the unit of protection, because it is more generally applicable to processes such as service sectors, administrative systems, etc. (Goldratt, 1990a:124). The concept of time versus inventory buffers is really the same thing, just viewed from different perspectives. When one uses time as the unit of measure for the buffers and prereleases work, physical inventory tends to accumulate at certain key locations in the manufacturing system.

Buffer Types, Checking Points, and Origins. TOC identifies three types of buffers. Constraint buffers "protect" the planned schedule of the constraint by positioning a certain amount of time, in the form of workload, in front of the capacity constraint before processing. If the buffer is inadequate, then the result is a loss in throughput to the plant. Shipping buffers protect due dates by ensuring that products are completed a certain amount of time before they are actually required to be shipped. Assembly buffers protect WIP that has already been processed on the constraint by ensuring that non-constraint WIP (not processed on any constraint resources) necessary for assembly with constraint WIP arrives at the assembly area a certain amount of time before it is actually required for assembly. The exact location of buffers is dependent upon the manufacturing system in question, but as a minimum, a linear system with a constraint will require a constraint buffer and a shipping buffer (Umble and Srikanth, 1990:144).

Since time is used as the unit of protection (for the buffers), there are no physical buffers, so one must designate a point where inventory tends to accumulate as a "buffer checking point" (Goldratt,

1990a:129). Information about the extent of Murphy is absolutely necessary for any reliable scheduling system; however it is impossible to measure the effect of Murphy on all individual resources. Instead, one must determine the aggregated effect of Murphy on the entire system (Goldratt, 1990a:119). Buffer checking points are important because they enable the tracking of the aggregated effect of all disturbances and they enable the buffers to be "attached" to the schedule, thus determining material release dates (Goldratt, 1990a:130). To determine the release date, DBR "attaches" the time buffer to the schedule of the future consumption of the constraint (Goldratt, 1990a:130). On the time axis, the schedule of consumption of the constraints is the origin of the time buffer, and the buffer stretches backwards in time (Goldratt, 1990a:130).

The physical locations from which the constraints consume the materials are called buffer-origins (Goldratt, 1990a:130). The types of buffer-origins correspond to each of the types of buffers: there is one for the resource buffer, the shipping buffer, and the assembly buffer. The buffer-origin of the resource buffer is located in front of the constraint and contains WIP to protect the constraint (Goldratt, 1990a:130). The buffer-origin of the shipping buffer is located at the shipping dock or the finished goods warehouse and it protects the market constraints (Goldratt, 1990a:130). The shipping buffer-origin either contains finished goods, or in the case where early shipments are authorized, it contains a list of orders that were shipped early (Goldratt, 1990a:130). The buffer-origin of the assembly buffer is placed only in front of non-constraint operations that feed assembly operations that are also fed by constraint resources, and it will only

contain non-constraint parts (Goldratt, 1990a:131). The use of these buffer-origins highlights the fact that TOC is very selective in what is to be buffered. Not every resource should be buffered--adequate protection can be achieved by buffering a relatively few critical locations.

Establishing Buffer Length. The stochastic nature of disturbances does not permit precise determination of the time buffer (Goldratt, 1990a:125). The dependent nature of resources further complicates the establishment of buffer protection since delays at one resource affect not only the output of that resource, but also other operations (Umble and Srikanth, 1990:143). Selection of buffer length involves a tradeoff: if one chooses a long buffer, due date performance will be very good, but lead time will be longer and inventory will be higher; if one chooses short buffers, the lead time and inventory will be reduced, but delivery performance will be poor and more expediting will be required (Goldratt, 1990a:126). Selection of a buffer length is often a judgement call. According to Schragenheim and Ronen, the initial length of the buffer is often based on "gut reaction," but it should be set at least three times the length of the average lead time to reach the buffer-origin (Schragenheim and Ronen, 1989b:20). Umble and Srikanth state that their experience has shown that a convenient starting point for the size of the total time buffer when implementing DBR is one half of the current lead time. This estimate not only provides a sufficient buffer for due date protection, but also it meets the need to minimize lead time. Lead times should actually be reduced since most of the time spent in current manufacturing systems is spent

in queue, and using DBR, the queues coincide only with the strategic placement of limited number of buffers (Umble and Srikanth, 1990:145).

Goldratt suggests that it is helpful to consider the probability of completing a task through many operations as a function of time (Goldratt, 1990a:125). The probability of overcoming a disturbance increases over time, but it never reaches certainty (100%) (Goldratt, 1990a:125). The decision on the length of the buffer should be made by the people responsible for the overall performance of the company. Experience in the operation is the best way to determine the proper buffer size: start with an estimate and adjust as you go (Chase and Aquilano, 1989:810). The bottom line is that the time buffer must be long enough to ensure that the bottleneck continues to work (Chase and Aquilano, 1989:809).

TOC provides a heuristic for examining the buffer length, using a "buffer profile." (Trigger, 1990:81). The buffer profile depicts the difference between the planned level of inventory (based on variation due to Murphy) and the actual WIP in the buffer-origin. To get a better picture of the profile of the difference between the planned and the actual WIP in the buffer-origin, TOC splits the buffer-origin into three zones. The first zone contains material about to be consumed by the resource, so it should be full or close to full. In contrast, one would expect to find much of the material planned for the last third of the buffer-origin (farthest from consumption), zone three, to be absent. The contents of middle third of the buffer-origin, zone 2, would be expected to lie somewhere between the two extremes. The buffer-sizing heuristic advocates adjusting the size of the time buffer until the above profile is achieved (Goldratt and Fox, 1986:122-27).

Selective Expediting, Tracking, and Process Improvement. The reader may now begin to recognize how TOC provides a focus for selective expediting, tracking, and process improvement efforts. Extending the heuristic described above, buffer management enables managers to examine the content of the buffer-origins to determine which tasks can be selectively expedited as follows (Goldratt, 1990a:135):

1. Pick a time buffer that is sufficiently long to permit a high percentage of tasks to arrive at the buffer-origin on time.

2. Check the buffer-origin to see if those tasks that should be there, with a given probability (say 90%), have arrived.

3. Expedite those jobs that have not arrived.

Selective expediting in this manner will considerably reduce the lead time of those tasks--they will not require as much time to arrive at the buffer-origin (Goldratt, 1990a:136). The "expediting zone" is analogous to region one in the above heuristic, with "holes" representing material that should have been in this zone but have not yet arrived. The partitioning of the three zones of the buffer-origin is established according to the percentage of tasks that management desires to expedite/track, calculated as the probability that the task should be in the buffer-origin subtracted from 1.0 (Goldratt, 1990a:136). Expediting is a trade-off between more inventory or more operating expense (management attention); however, planned expediting definitely requires much less OE than unplanned expediting.

To make the procedure more reliable, buffer management also includes a lesser percentage of items to track only, without expediting. For example, if one expedites those tasks that should be in the buffer-

origin with 90% probability but are not, then one might want to simply track tasks that have a 60% probability of being in the buffer-origin, but have not yet arrived (Goldratt, 1990a:140). This "tracking zone" corresponds with zone 2 from the buffer-sizing heuristic described above. If the system relies exclusively on expediting (i.e., at the 90% threshold), then problems will often not be detected until the tasks have moved to subsequent resources (Goldratt, 1990a:140). One should refrain, however, from attempting to track too much. Goldratt asserts that attempts to track at greater than the 60% level will cause the system to become too cumbersome (Goldratt, 1990a:140).

The use of buffer management in this manner provides a systematic approach not only for selective expediting and reduction of the buffer size, but also for focusing process improvement efforts. Once the late tasks are identified, the first action is to determine where they are stuck. By recording where each of the late tasks resides, one can develop a list of problematic resources, some of which will appear multiple times (Goldratt, 1990a:138). Use of tracking will not only identify the problems earlier, but also it will improve the reliability of the problematic resource list (Goldratt, 1990a:140). Using this list to deal with problems at the resource (versus the task) level, management can pinpoint common problems--they can see which resources are causing problems over multiple tasks (Goldratt, 1990a:139). This practice allows management to concentrate on increasing throughput rather than continually "fighting fires" (Trigger, 1990:83).

Establishing Control: The Rope. Once the drum has been authorized and appropriate buffers have been established to protect throughput, communication must be provided to ensure that all resources

work in accordance with the drum. The planned production at each resource is tied directly to the drum through the use what TOC calls a "rope." The rope enables synchronization of all non-constraints without having to actively control each and every resource (Umble and Srikanth, 1990:138). It ensures that the proper amount of material is released into the system at the right time and communicates the actions required to support the MPS throughout the plant (Umble and Srikanth, 1990:138).

In most plants the majority of material flow problems are the result of overactivation of non-constraint resources (Umble and Srikanth, 1990:165). Communication from the "rope" ensures that all resources work up to, but not in excess of, the rate of the drum by backward scheduling from the buffer the release of all materials into the shop floor (Schragenheim and Ronen, 1989a:4). The simple rule in DBR is for non-constraints to work only on material that is available at the work center. Although specific schedules are not produced for the non-constraints, the release of raw materials based on the drum determines the activity of non-constraint resources.

DBR attempts to control only certain key points in the productive system, called "schedule release points." Schedule release points are points in the product flow where a detailed schedule is necessary to maintain control. A detailed schedule is necessary whenever sufficient material is not available to permit work at the next work center (Umble and Srikanth, 1990:165). According to Umble and Srikanth, schedule release points are limited to four circumstances: 1) material release points, 2) constraint resources, 3) divergence points, and 4) assembly points (Umble and Srikanth, 1990:166). Since the release of material is a key to controlling the plant, gateway operations must be strictly

controlled in any manufacturing operation. Effective control of constraints is also essential, so these resources are also schedule release points. Divergence points are points in the material flow where a particular material can be processed into different products. These points must be schedule release points to avoid misallocation of material--workers at these points must know how much of each product to produce and the priority sequence of the products. Finally, assembly operations also require detailed schedules to ensure that all required parts are available for assembly.

Chapter Summary

Thus far, the research has reviewed MRP-based systems, including AFLC's plans for DMMIS. In addition, JIT was briefly introduced. This chapter culminated with a discussion of TOC, particularly with respect to its drum-buffer-rope method of scheduling. All of this information leads to the discussion of *DISASTER*™. Sections on MRP and JIT provide a means for comparing *DISASTER*™ to other commonly-used approaches. The TOC section provides the essential basics necessary to understand the logic behind *DISASTER*™. Next, the research will identify the methodology employed for this research, followed by a review of the characteristics and logic of *DISASTER*™.

III. Methodology

Explanation of Research Method

The particular method chosen for this research includes a combination of the historical method, an in-depth review of the operation of the software itself, and a single, holistic case analysis of a company implementing *DISASTER™*. Historical research, according to Borg and Gall, is "a systematic and objective location, evaluation, and synthesis of evidence in order to establish facts and draw conclusions concerning past events" (Borg and Gall, 1971:260). The objective of the software review is to document *DISASTER™*'s internal processing logic. A case study is an empirical inquiry that investigates a contemporary phenomenon within a real-life context using multiple sources of evidence (Yin, 1984:23).

The Literature Review. Private industry has developed (and is now applying) an enormous knowledge base of manufacturing technology. The objective of the literature review is to provide enough information on traditional and leading-edge manufacturing philosophies, including AFLC's Depot Maintenance Management Information System, to permit a knowledgeable review of the *DISASTER™* system. In line with these objectives, the review briefly explores materials requirements planning (MRP), manufacturing resource planning (MRP II), and DMMIS. Next, the review introduces relevant aspects of just-in-time (JIT) manufacturing. Finally, the review discusses concepts of the theory of constraints (TOC) that form the basis for scheduling by *DISASTER™*. The researcher obtained general information for the literature review from trade journals, periodicals, books, and popular magazines, as well as

conference proceedings, student papers, theses, and dissertations. Additional data specific to DMMIS was obtained from Air Force correspondence, briefings, and bulletins.

The Software Analysis. To date, no one has performed an analysis of the *DISASTER*™ program. Using a copy of the *DISASTER*™ software, the documentation, and data from the literature review, the researcher will perform such an analysis. The researcher obtained a complete copy of the scheduling block of the *DISASTER*™ program. The general intent of this analysis is to reveal what is required for *DISASTER*™ to produce reliable, realistic schedules and to investigate precisely how *DISASTER*™ operates. The basic components of this analysis will include descriptions of hardware requirements, necessary inputs, processing logic, outputs, and anticipated benefits.

The Case Analysis. A case study is similar to an historical review; however, it adds two additional sources of information: direct observation and systematic interviews (Yin, 1984:19). A holistic design involves examination of only one unit, often at a higher level of analysis (Yin, 1984:44). A major strength of case study research is the opportunity to use multiple sources of data (Yin, 1984:90). A case analysis "represents an intensive study of phenomenon using a variety of data sources and tools" (Borg and Gall, 1971:84). Unfortunately, researchers using the case analysis method normally "have no standard procedure to follow and must be flexible and attempt to glean information and insights from wherever they may find them" (Zikmund, 1988:84).

The general analytic strategy for this case study is descriptive--to document the use of *DISASTER*™ in a real-world environment. The

company selected for this study is *The Zycon Corporation*, a circuit board manufacturer who is one of the leading advocates of the use of TOC-based manufacturing. The researcher collected data for the analysis on-site (in Santa Clara, California) through observation and informal interviews.

Methodology Justification

Before one can understand the requirements and potential benefits regarding use of *DISASTER*™, one must understand traditional manufacturing planning and control systems in general, and TOC in particular. The literature review is essential for providing this required knowledge base. According to Lang and Heiss:

> through history one can develop a background perspective and insight into a person, problem, event or institution not obtainable through other types of research. In historical research, which is often most concerned with qualitative results, the historical methodology does generate the answers. (Land and Heiss, 1984:65)

This research is somewhat exploratory in nature, and is intended to "pave the way" for future research that can experiment with actual performance of the program.

According to Yin, the case study research design is preferred when examining contemporary events, especially when relevant behaviors cannot be manipulated by the researcher (Yin, 1984:19). Furthermore, selection of a single case study design is completely justifiable when the case represents a "critical test of existing theory, where the case is a rare or unique event, or where the case serves a revelatory purpose" (Yin, 1984:47). *DISASTER*™ was only recently released (February 1991), and no companies have completed implementation of the system. *The Zycon Corporation* appears to have made the most progress in this area. Zycon's efforts represent a unique, first-time opportunity to

investigate operation of the software; therefore, selection of the single holistic case study design was appropriate. Use of this case analysis to supplement the literature review and software analysis will provide a level of external validity that would not be possible with a literature review and software analysis alone.

IV: Analysis of DISASTER™

Introduction and Scope

The scheduling phase of *DISASTER™* was originally intended only to be a software tool to support DBR and serve as a first step in implementing an information system based on the throughput world; however, much faster processing times were achieved than originally intended, permitting additional capabilities to be included in this phase (Avraham Y. Goldratt Institute, 1990d:2). One of the most significant improvements is the capability to simulate future schedules, allowing conflicts to be resolved prior to release of the schedules to the shop floor (Avraham Y. Goldratt Institute, 1990d:2). When schedules are released, they are more immune to disturbances on the shop floor, which for lack of a more descriptive term will be referred to as "Murphy" (Avraham Y. Goldratt Institute, 1990d:2).

Goldratt's latest book, *The Haystack Syndrome*, was distributed with the software to explain the conceptual basis for *DISASTER™*'s scheduling logic; therefore, many of the ideas presented in this section are cited directly from this book. Much of the remaining discussion is based on the *DISASTER™* documentation. The review of TOC concepts in Chapter 2, especially with respect to drum-buffer-rope scheduling, also applies directly to the discussion of *DISASTER™*'s processing logic.

The objective of this section is to review the software package. The purpose of reviewing the software is to examine conceptually how *DISASTER™* operates and to discuss the general characteristics of the program. This review will identify program requirements, the logic

behind *DISASTER*™'s scheduling procedure, and anticipated benefits that might be realized through its use.

Preliminary Concepts and Foundation

The Information Systems Hierarchy. According to TOC, any information system must perform three major functions: scheduling, control, and what-if analysis (Avraham Y. Goldratt Institute, 1990d:4). The most basic stage of the *DISASTER*™ information system is the schedule stage. Scheduling involves determining who should do what, when, and in what quantities--it is a list of answers to managerial questions. Generation of reliable schedules is the building block of any information system and it is a prerequisite to performing any what-if analyses (Goldratt, 1990a:117-8). To provide realistic schedules, the *DISASTER*™ schedule block requires rough estimates of the time buffers and of the required levels of protective capacity (these estimates are later refined by the control block) (Goldratt, 1990a:158). In addition, the schedule block must recognize and account for disturbances (Goldratt, 1990a:159).

The control stage involves estimation of the impact of disturbances on the shop floor: getting a handle on the tradeoff between protective inventory and protective capacity (Goldratt, 1990a:158). This stage identifies areas on which to focus process improvement efforts and defines how to handle local performance measurements (Goldratt, 1990a:158). Development of reliable schedules by the scheduling stage is mandatory for the proper execution of the control stage.

The ultimate goal of *DISASTER*™ is the ability to perform what-if analyses. This stage is geared to elevating constraints and preventing

the creation of unnecessary constraints. Typical analyses involve decisions such as investment strategy, make or buy, or product mix. The what-if block of the information system cannot operate unless the schedule and control blocks already exist (or at least will be corrected if identified). Since this stage is dependent upon the previous stages, what-if analysis can only be used reliably after the constraints have been identified, Murphy has been quantified, and realistic levels of protective capacity have been established (Goldratt, 1990a:117,159).

Data versus Information. Two general conditions must be present for *DISASTER*™ to supply good information: data and a decision process (Goldratt, 1990a:81). Goldratt makes a point of distinguishing between the commonly-confused terms *data* and *information.* Data is any string of characters that describes anything about reality while information is data that has been formatted to answer specific questions. Goldratt defines information as "the answer to the question asked" (Goldratt, 1990a:6,156). Information is not readily available, but must be deduced from the required data. The nature of an information system is different from a data system. An information system should contain information in a hierarchical structure such that higher level information can be deduced (through a decision process) from the lower levels. Furthermore, what is information at one level may be only data at another level. An information system should draw required data from a data system. In contrast, a data system is geared to answer straightforward questions outright, without the requirement for a decision process.

The Required Decision Process. The throughput world requires a new decision process to bridge the gap between data and information

(Goldratt, 1990a:103). The five focusing steps discussed under the DBR section provide the basic decision process for *DISASTER™*. These steps enable *DISASTER™* to move up the information ladder from basic data to higher levels, and eventually to the financial bottom line (Goldratt, 1990a:82). As with manual applications, identification of the constraints is not done in one step, but rather is done through an iterative process. Since the number of constraints will always be small, the iterative nature of the process does not present a big obstacle (Goldratt, 1990a:117). Some constraints can only be found after the exploitation step, while others can only be found after the subordination step (Goldratt, 1990a:117). Since some constraints can only be identified *after* the exploit and subordination steps, then the only way to identify all the constraints without real life iterations is to simulate future events (Goldratt, 1990a:117). It follows that in order to identify even the current constraints, *DISASTER™* must simulate future events. This fact highlights the need for scheduling in any information system (Goldratt, 1990a:117).

The five focusing steps do not provide sufficient guidance for *DISASTER™*. The system also requires detailed procedures that relate to each step and can be applied generally. First, *DISASTER™* must identify and exploit the constraints, and subordinate everything else to them. Goldratt stresses that it would be very difficult (if at all possible) for an information system to identify policy constraints, so *DISASTER™* assumes none exist. Next, it must determine the noise level (the level of disturbances), based on actual daily transactions, and "translate" this noise into the proper amount of safety inventory. These safety

buffers then need to be properly placed to minimize the impact of Murphy (Goldratt, 1990a:111).

The Need for Knowledge. *DISASTER™* is not a typical software package--its basis is difficult for most cost-world managers to understand and use (Avraham Y. Goldratt Institute, 1990d:1). As noted, the more thoroughly the user understands the principles of the TOC, the more benefits he or she may reap from its use. Only if the users understand the underlying principles upon which *DISASTER™* is based will they be able to realize the program's full potential (Avraham Y. Goldratt Institute, 1990d:1). Furthermore, although it can be a very powerful tool, if used without the skills and knowledge necessary to control the power of the package, *DISASTER™* can lead to "disaster" (Avraham Y. Goldratt Institute, 1990d:1). For example, the software intentionally disregards policy constraints. If *DISASTER™* is used in an environment that contains policy constraints, the program will not achieve its full potential. *DISASTER™* is intended for use only in the throughput world, where policy constraints have been (or will be) removed (Avraham Y. Goldratt Institute, 1990d:6). Goldratt stresses that *DISASTER™* should only be implemented in organizations that have already undergone the necessary change to the throughput world (Avraham Y. Goldratt Institute, 1990d:6).

The Criteria for a Good Schedule. TOC stresses that any schedule must be realistic, and to be realistic the schedule must 1) recognize the constraints (those things that limit performance) and identify conflicts between the constraints, and 2) have a certain amount of immunity against possible disruptions (Goldratt, 1990a:180). When *DISASTER™* goes through the process of identify-exploit-subordinate,

each time a new constraint is identified, the system must check for conflicts between the constraints (Goldratt, 1990a:181). Since the system does not have all the intuitive information necessary to resolve conflicts between the constraints, it does not attempt to do so. Instead, it concentrates on identifying the conflicts and presenting the user with possible actions that can be undertaken to resolve them (Goldratt, 1990a:181). The system is not intended to contain *all* data be included in the system (i.e., not the intuitive data), but instead it leaves all judgmental decisions to the user (Goldratt, 1990a:185). Based on this criterion, *DISASTER*™ only permits inventory that protects throughput (any additional is excess) and OE (i.e., overtime) that is preauthorized and necessary to protect throughput. Any additional actions regarding use of inventory and OE are left strictly to the user (Goldratt, 1990a:185).

To illustrate potential conflicts between constraints, Goldratt presents an example of a case where salespeople have promised a delivery date that cannot be met by normal processing of the company's resources (Goldratt, 1990a:181). In this case, there is a conflict between the market constraint and the resource constraint(s). Either the user can dump more overtime on the resources to speed up processing, or the delivery date can be postponed. In either case, the user must make the decision.

With respect to the second requirement for realism, immunity, if any disturbance necessitates the generation of a new schedule, then the original schedule was not realistic (Goldratt, 1990a:183). Since scheduling must identify future constraints, then the schedule must be realistic over this future time period (Goldratt, 1990a:183). Both MRP

and JIT recognize the significance of disruptions to the system. In fact, they actually allow more time for protection against disruptions than for the processing itself. Goldratt claims that the problem with these methods is that both have tried to immunize the schedule itself, rather than the result of the schedule--they have tried to protect each and every individual process, resulting in an over-extended lead time (Goldratt, 1990a:183). In addition, these methods, especially JIT, rely excessively on floor personnel to resolve scheduling issues (Goldratt, 1990a:184).

Scheduling Procedure/Logic. The procedure used by *DISASTER™* follows the same general guidelines (i.e., the focusing steps) discussed under the section on DBR scheduling. First, the system constraints are identified and exploited (authorizing the drum), then all other resources are subordinated to the decision about the drum. This section provides more detail specific to *DISASTER™*'s use of TOC concepts to produce realistic schedules.

Identification of the Constraints. As with standard DBR scheduling, the first place to begin in scheduling is to identify a resource that is *definitely* a constraint. If there is any doubt, the resource should be declared as a non-constraint, since if it really was a constraint, it will be identified by a subsequent iteration (Goldratt, 1990a:187). It is usually too risky to start with either a vendor or a material constraint since one has limited information about the schedule at this point (Goldratt, 1990a:187). Starting with a resource constraint is also risky since most of them are nonbottlenecks. Again, it is difficult to identify a true resource constraint unless you already have the schedule (Goldratt, 1990a:187). Since the market is

always a constraint when due-dates are specified, then the market constraint is the logical place to begin developing the schedule (Goldratt, 1990a:198).

To identify bottleneck resources at this early stage in schedule development, the user must first provide a cut-off date, called the schedule horizon (Goldratt, 1990a:191). Using the schedule horizon, *DISASTER*™ then determines the total load placed on each resource during the horizon (Goldratt, 1990a:191). In calculating these loads, *DISASTER*™ uses all tasks whose due-date is before the schedule horizon; however, even those tasks that are due only a few days after the cut-of date (schedule horizon) still place a load on resources during the scheduling horizon. To account for these tasks, *DISASTER*™ includes in the load calculations any order whose due date is less than the scheduling horizon *plus* the shipping buffer (Goldratt, 1990a:192). Calculating loads in this manner ensures that *DISASTER*™ recognizes all relevant processing during the schedule horizon (Goldratt, 1990a:192). At this stage, *DISASTER*™ also accounts for setup time, but it only permits the minimum number of setups: one per operation (Goldratt, 1990a:192). TOC recognizes that setup time can be saved by combining batches; however, it is not preferable for the system to include setup time savings during identification of the bottlenecks (Goldratt, 1990a:193). Setups are considered later, as a last resort.

Once the load per resource for each resource type is calculated, then the availability of the resource over the scheduling horizon (not including the shipping buffer) is calculated and the figures are compared for each resource (Goldratt, 1990a:193). If the load placed on any resource (or group of resources) is greater than its availability,

then the system contains at least one bottleneck (Goldratt, 1990a:193).

In some cases, the above comparison will reveal that more than one resource appears to be a bottleneck--it has less capacity than required (Goldratt, 1990a:194). In such cases, not all the suspect resources are declared as constraints. Only the one that limits throughput the most should be designated as a *likely* constraint (Goldratt, 1990a:195). The constraint is not immediately labeled as a *definite* constraint due to the likelihood of data inaccuracy, so data must be verified at this point (Goldratt, 1990a:194).

Verifying the Data. To verify the data, the number of resource units available should first be checked. Goldratt notes that, surprisingly, this number is often incorrect. Next, the process times used to calculate the loads are checked. Not all times are checked, just the ones for tasks necessary to fulfill an order for which there is at least one demand within the schedule horizon (Goldratt, 1990a:197). Even then, not all of the task process times are equally important. The important ones are those that are much longer or must be done multiple times (Goldratt, 1990a:197). Unlike MRP systems, *DISASTER™* clearly highlights which data elements should be checked, and presents this information to the user (Goldratt, 1990a:197). *DISASTER™* provides a chart that identifies how much (the percentage) of the suspected constraint's availability each task absorbs (Goldratt, 1990a:197). Normally, the above chart will reveal that only a few tasks create the majority of the load on the suspected constraint. These "big-load tasks" are the process times that need to be verified (Goldratt, 1990a:197).

In addition to checking process times for the "big-load tasks," *DISASTER™* also enables the user to check the accuracy of orders that require these tasks (Goldratt, 1990a:198). This information could be very voluminous, so *DISASTER™* does not attempt to store all data as it goes (the present MRP method), but rather it "implodes" from the suspected constraint back to the market constraint and marks the path with "red lanes" (Goldratt, 1990a:198). Once the path has been marked, since all the required data to redo the calculations is stored in memory, *DISASTER™* can recalculate (rather than retrieve!) the information. It implodes to identify all tasks that require the particular resource, and then does a selective explosion (Goldratt, 1990a:198). During the implosion, only the relevant load data is stored to memory, so the memory requirements are relatively small (Goldratt, 1990a:199). If the user wants to see specific, detailed order data, then he or she has the option of requesting that additional data also be stored during the explosion process (Goldratt, 1990a:199). Since all the required implosions and explosions are done in random access memory (RAM), the time required for this procedure is minimal (Goldratt, 1990a:199).

Identifying and Resolving Conflicts. The next step of *DISASTER™*'s scheduling process is resolution of apparent conflicts between the constraints (Goldratt, 1990a:199). Subordinating resources means that a resource does not try to produce to maximize its own potential, but instead it does exactly what is needed to satisfy the constraint (Goldratt, i990a:201). To determine the appropriate number of units needed to "satisfy" the constraint, *DISASTER™* first explodes the product structure (Goldratt, 1990a:202). In addition to exploding,

the system must also allocate existing stocks to the various orders/tasks (Goldratt, 1990a:202). Rather than using a sophisticated decision rule, *DISASTER™* simply allocates according to first come, first served (FCFS): early due-dates win over later due dates (Goldratt, 1990a:202). Using a simple allocation rule such as FCFS avoids the necessity of *DISASTER™* needing "intuitive" information regarding the priority of the various orders.

Now that *DISASTER™* has calculated the number of units required for a particular resource, and the resulting load has been calculated, it now "places" this load on the time axis to identify when it should be processed relative to other loads (Goldratt, 1990a:203). Given the due date and the length of the shipping buffer, the resource constraint should complete its processing on each load a shipping buffer before it is due, so *DISASTER™* simply places it on the time axis accordingly (assuming no disturbances) (Goldratt, 1990a:203). This same procedure is performed on all tasks that require work on the constraint resource: each task is placed on the time axis so that it will be completed a shipping buffer before it is due (Goldratt, 1990a:203). To ensure that any allocation of existing stock is made according to due-date, *DISASTER™* starts with the earliest due-date, and allocates any existing stock as it goes. For each order for which there is insufficient existing stock, a block is placed on the resource's time axis (Goldratt, 1990a:204). Note that up to this point *DISASTER™* has not considered the resource's limitations, so the resulting picture resembles a "ruins" similar to the one shown in Figure 5 (Goldratt, 1990a:204).

The next step is to "level the ruins" by making sure that only one block exists for each available resource (Goldratt, 1990a:204).

DISASTER™ levels the ruins by pushing all orders above the available number of resource units to the left (backward in time), making their processing earlier than strictly demanded by the due date (Goldratt, 1990a:204). The key to leveling the ruins is to ensure that any block-load that was originally due to be completed before another block-load is still completed before the subsequent block-load (Goldratt, 1990a:205).

After leveling, some blocks will undoubtedly be moved beyond the present and into the past (Goldratt, 1990a:205). To correct this situation (you cannot instruct resources to work yesterday), *DISASTER™* now pushes all jobs to the right (forward in time), maintaining each job's position relative to other jobs, until the first job begins at the present (Goldratt, 1990a:206). This method of arranging the blocks does not eliminate late deliveries, but instead it clearly exposes problems (Goldratt, 1990a:205). All of this shifting will leave the blocks in a different position than where they were originally placed, and since this resource is a constraint, then some of the blocks will definitely wind up farther to the right than originally placed: they will not be completed a shipping buffer early (Goldratt, 1990a:206). If a block is now positioned such that it is being completed less than half the shipping buffer before its due date, then there is a high probability that Murphy will cause it to miss its due date, so *DISASTER™* colors these blocks red (Goldratt, 1990a:206).

DISASTER™ now checks to ensure that, for each of the scheduled blocks, all of the material required for that operation is available (Goldratt, 1990a:209). The system must ensure that the first blocks have the necessary material available. If not, *DISASTER™* must

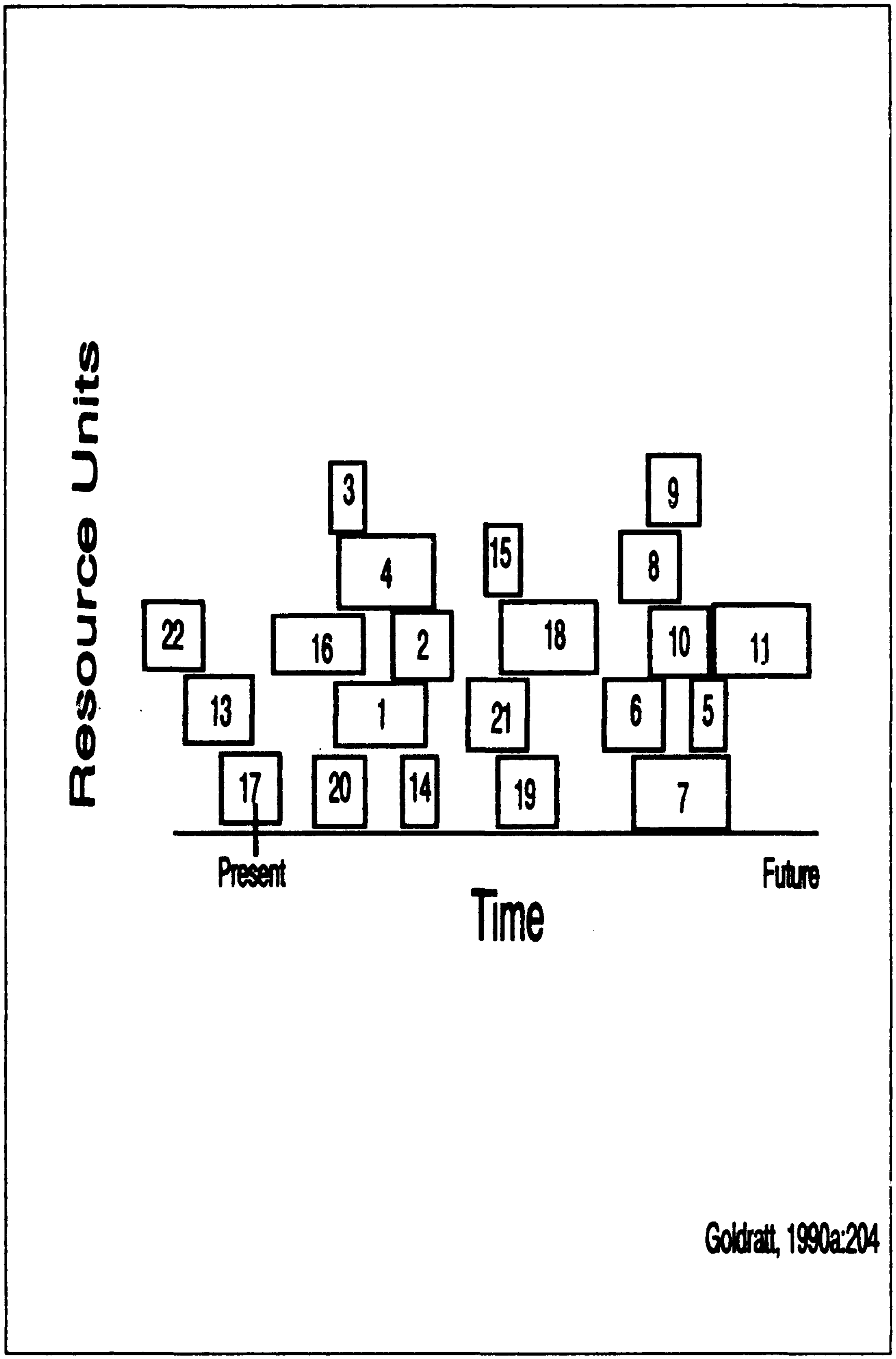

Figure 5. The Ruins

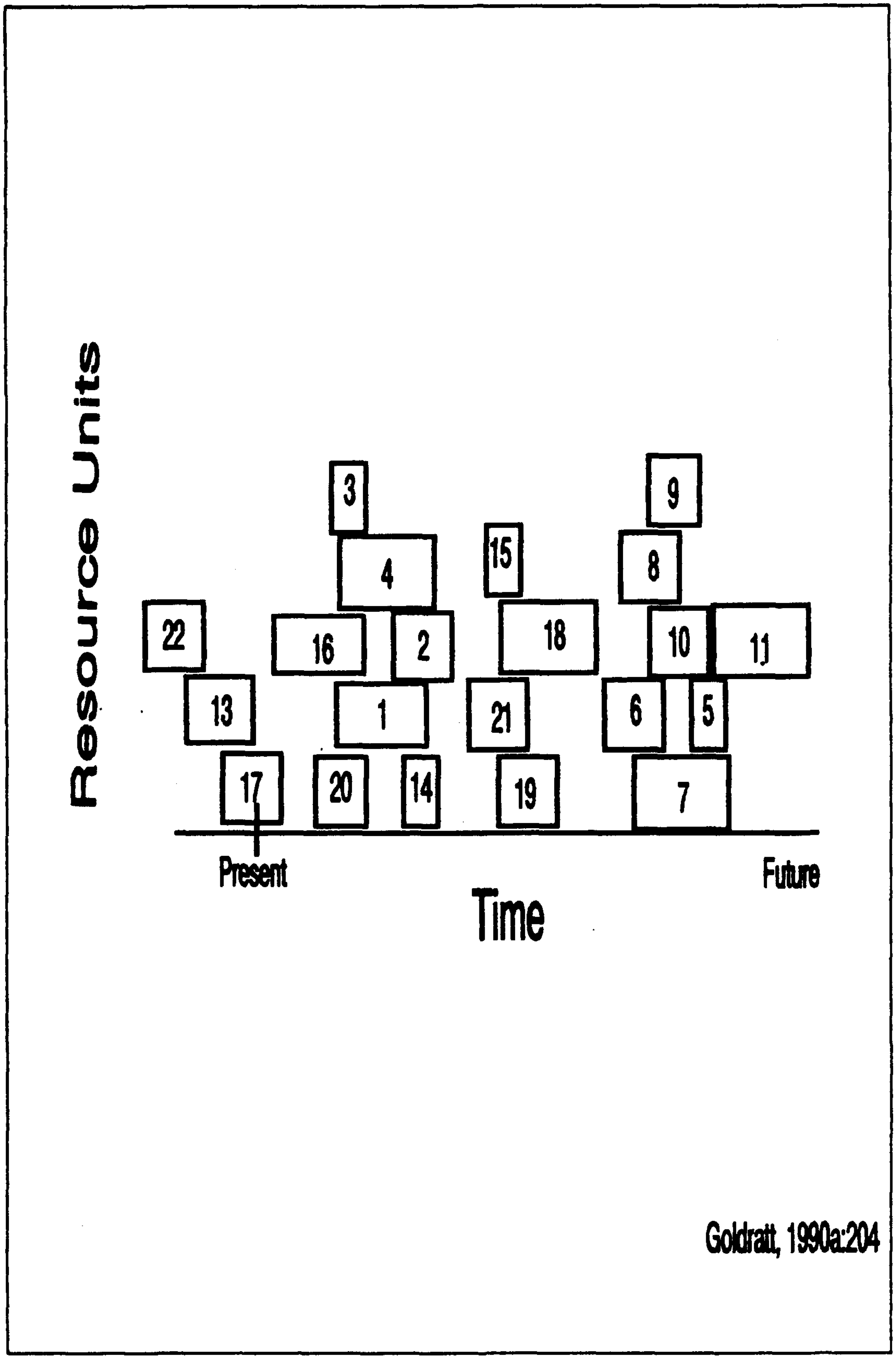

Figure 5. The Ruins

resequence so that the earliest required blocks that have all the necessary material are at the beginning (Goldratt, 1990a:209). *DISASTER™* uses a percentage of the resource constraint buffer (the time needed for a task to go from release to the resource constraint with a very high probability of on-time arrival) as the amount of time to move the operation during this resequencing (Goldratt, 1990a:209).

Exploiting the Constraint: Setups and Overtime. At this point, *DISASTER™* now returns control to the user for direction on where savings in setup should be made (Goldratt, 1990a:210). Since setup savings do not follow any generic pattern (i.e., order due-dates), *DISASTER™* presents the user with a picture of the potential setup actions and asks him or her to make the decision (Goldratt, 1990a:210). Each time more than one block of the same task is to be performed, the possibility exists to perform the blocks consecutively (to "glue" them together) and thus save a setup. If any in-between blocks are red, and *DISASTER™* automatically performed the gluing, then the program would be assuming that the glued block is more "important" than the in-between block (Goldratt, 1990a:212). Since *DISASTER™* is not intended to make judgmental decisions, control must be turned over to the user to enable him or her to make the decision (Goldratt, 1990a:212).

Any saved setup frees more constraint capacity, facilitating exploitation of the constraint; however, setups also present a problem: to "glue" blocks together, one block will have to be processed earlier than otherwise required, thus it will have to skip over other blocks (Goldratt, 1990a:211). Any orders that were in between will be processed later, delaying their completion (Goldratt, 1990a:211). In addition, inventory must be released to the operation faster, resulting

in more WIP (Goldratt, 1990a:211). Setup savings are only practical if red blocks exist after the second block. Otherwise, no improvement is possible and the savings would not be beneficial (Goldratt, 1990a:211). When two blocks are glued, all blocks that were located after the moved block will gain: the blocks will be processed earlier by a time equal to the setup time saved; however, all in-between blocks will lose: they will be done later by a time equal to the processing time of the second glued block (Goldratt, 1990a:212).

Since different tasks require different setup times (even when done on the same resource), different tasks represent different opportunities for setup savings (Goldratt, 1990a:213). Also, the farther the system reaches into the future to pull blocks back for gluing, the higher the cost--more blocks will have to be skipped (Goldratt, 1990a:212). Based on these observations, *DISASTER™* limits the maximum time that blocks can be pulled back depending upon the amount of setup savings possible (Goldratt, 1990a:213). *DISASTER™* requests that the user provide a distance/savings ratio, and it gives him or her the capability to try several different ratios before deciding on the appropriate one (Goldratt, 1990a:213). This ratio provides *DISASTER™* with general gluing instructions, preventing the user from having to proceed block-by-block (Goldratt, 1990a:212).

At this point, *DISASTER™* now turns its attention to overtime. The user is allowed to preauthorize overtime using broad guidelines by day, week, and weekend. *DISASTER™* then automatically assigns overtime within these limits (Goldratt, 1990a:214). For the non-constraints, *DISASTER™* does not permit overtime assignment in addition to the preauthorized levels; however, if setup savings on the constraint are

not sufficient, additional overtime may be required (Goldratt, 1990a:214). If there are no red blocks, then there is no need for overtime.

Use of overtime will help not only the specific block intended, but also any block that was scheduled for processing after this block (Goldratt, 1990a:215). *DISASTER™* always places overtime before the first red block (otherwise its use would not prevent a late delivery); however, the earlier the overtime is assigned relative to the time the block is scheduled for processing, the more the system pays in increased inventory (Goldratt, 1990a:215).

To assign overtime, *DISASTER™* starts at the earliest red block and moves backward in time, allocating any authorized overtime until either the block in question is no longer red or it encounters the present. If the present is reached and the block is still red, then it is time to return control to the user for additional guidance and options (Goldratt, 1990a:216). Once this first red block is dealt with, then *DISASTER™* moves to the next red block and performs the same procedure (Goldratt, 1990a:216). After all the red blocks have been accounted for, then the user will still have the option of trying to off-load, split the block, use more overtime, or use some other means to correct the situation (Goldratt, 1990a:216). Even though all actions are, at least initially, aimed at a particular block, when overtime is allocated, its use improves the entire system (Goldratt, 1990a:217).

Before quitting, *DISASTER™* presents a screen with the new status of all blocks (Goldratt, 1990a:217). At this stage the user can see the impact of all previous decisions, and he or she is allowed to go back and make any desired changes (Goldratt, 1990a:217). When the user is

satisfied that no more changes can (or should) be made, then he or she can stop and inform the customer that the due date will not be met (Goldratt, 1990a:217). Only after this procedure is completed does the user have a first attempt at a realistic master production schedule (Goldratt, 1990a:217).

Subordinating in Accordance with the Drum. The next step in *DISASTER™*'s scheduling procedure is to subordinate all actions in accordance with the MPS developed for the drum (Goldratt, 1990a:218). In some cases, a particular order may be processed on the constraint more than once. These situations present a problem: how can *DISASTER™* determine the appropriate levels of protection for both constraint operations? The system must ensure that the second operation is not starved (that would affect throughput), while simultaneously minimizing the length of the buffer (or the company will pay in extra inventory and a longer lead time) (Goldratt, 1990a:218). *DISASTER™* applies the concept of "rods" to solve this problem. The use of "rods" ensures that at least half the resource buffer is always between the two blocks (Goldratt, 1990a:218). Some blocks can have multiple rods, so their movement affects more than one block (Goldratt, 1990a:219). In reality, the blocks are an interval rather than a point estimate of time, so *DISASTER™* also considers the length of the two blocks when determining the length of the rods. In all cases, the system must ensure that each unit completed at the early block will have at least half a resource buffer before it arrives for processing on the later block (Goldratt, 1990a:220). If the first process is much longer, then *DISASTER™* uses the last unit to determine the rod length, and if the first block is

much shorter, than it uses the first unit to determine rod length (Goldratt, 1990a:220).

Up to this point, only the logic with respect to operations of the drum, order due dates and resource constraints, has been discussed (Goldratt, 1990a:222). The next step is to consider the non-constraints--to subordinate them to the scheduling decisions of the drum (Goldratt, 1990a:222). Determining the material release date for non-constraints is straightforward. *DISASTER™* simply subtracts the shipping buffer from the order due-date (Goldratt, 1990a:222). Operations at the non-constraints will then proceed whenever material for the operation arrives, just like any DBR system. In fact, rather than scheduling when the non-constraint resources *should* work, *DISASTER™* really only schedules when they should *not* work. A non-constraint resource should never work before it is intended to do so, since this work will only result in excess inventory. The circumstances when a particular schedule needs to be specified for non-constraints were identified in the DBR scheduling section: schedules are required for any of the schedule release points.

The most obvious place for conflicts resulting from *DISASTER™*'s scheduling procedure is for material requests to be pushed back farther than the present (Goldratt, 1990a:228). The key to resolving this conflict is protective capacity (Goldratt, 1990a:230). *DISASTER™* identifies when peak loads will occur, then tries to smooth these peaks by shifting them to *previous* times when the resource in question is not overloaded (Goldratt, 1990a:230). To assist in this objective, *DISASTER™* determines daily resource loads by looking at the daily resource availability (from the calendar), and comparing this figure to

projected daily requirements (Goldratt, 1990a:230). Using this approximation, and the fact that no resource can be loaded at greater than 100%, *DISASTER™* then highlights times when loads need to be redistributed (Goldratt, 1990a:231). Since subordination is dealing with non-constraints, overtime use is not appropriate (Goldratt, 1990a:231).

The loads must be moved to some point back in time, otherwise they will not meet the due-date (Goldratt, 1990a:232). Since *DISASTER™* must move them back in time, it needs to consider *future* load profiles *before* the schedule is actually developed--otherwise, capacity limitations will be considered too late, after the schedule has already been impacted (Goldratt, 1990a:232). If the resource is loaded too much for the present date, then the only option is to delay processing to the next day. *DISASTER™* only schedules an operation after all of the following operations have been scheduled. Using this technique, *DISASTER™* is always assured that when it moves the peak loads backward, they will be placed on days with available capacity (Goldratt, 1990a:232).

In the manner described above, *DISASTER™* considers capacity limitations concurrent with schedule development (Goldratt, 1990a:232). In scheduling the non-constraints during subordination, *DISASTER™*'s logic requires that it begin with the *latest* order, raising the question of how to allocate stock (you do not want to issue it to the latest orders!) (Goldratt, 1990a:235). Recall that stock for red-lane processes has already been allocated prior to authorization of the drum (Goldratt, 1990a:235). Furthermore, the user has likely stopped at some point and postponed some of the due-dates (Goldratt, 1990a:235). Since the order of the blocks has now been set, *DISASTER™* now allocates all

existing stock to the remaining resources exactly as it did before: by order due-date (Goldratt, 1990a:235).

At this point *DISASTER*™ is now ready to "shovel" the peak loads. Redistributing the peaks is necessary because, due to capacity limitations, processing is not possible at the desired time (Goldratt, 1990a:236). Required capacity is determined largely by the buffer length (Goldratt, 1990a:236). There is a trade-off involved with moving the peak load earlier versus later. As it is moved farther backward, earlier material release dates are required, increasing the buffer time and the inventory. As the load is moved forward, block completion is delayed (Goldratt, 1990a:236).

The most important guideline during subordination is to move consistently back in time (Goldratt, 1990a:247). While *DISASTER*™ moves through the processes, it never dives into operations that are to be done earlier. Instead it records a note in a reminder file (prioritized by date) to deal with any related operations when it reaches that date (Goldratt, 1990a:248). When scheduling the operations, three conditions might be encountered that require such reminders: an activity of the drum, the buffers, and peak loads (Goldratt, 1990a:248). At the start of subordination, the reminder list is not empty. It includes all the drum operations that have been previously specified: the order due-dates and the ending time of the resource constraints (Goldratt, 1990a:249).

During the subordination process, *DISASTER*™ starts with the latest event (the order) and dives into its feeding operations. Before it can dive into feeding operations, it must first subtract the shipping buffer, and since *DISASTER*™ does not want to move from the current date, it just records the date for this feeding operation in its

reminder file and moves to the next operation. The program proceeds until all operations that directly feed the order are identified (and reminders are recorded). Whenever a new entry is added to the reminder list, it is positioned according to date assigned, and any calculated date that is before the present is set equal to the present. After all of the feeding operations are dealt with, the order is erased from the reminder list and *DISASTER™* moves to the next event on the list (Goldratt, 1990a:249).

Eventually, *DISASTER™* will encounter an operation, not an order. For each operation, the program calculates the load that must be placed on the required resource and adjusts the available capacity of the resource accordingly (Goldratt, 1990a:249). If the additional load is greater than the available capacity, then any surplus load is placed into a "left-over load entry" for that particular resource (Goldratt, 1990a:249). Once a resource's capacity is exhausted, then *DISASTER™* will not schedule any more operations on that resource unless the current date reaches the present date. When this occurs, all loads are temporarily placed on the current date (Goldratt, 1990a:250).

DISASTER™ continues diving through the operations until it meets one of three conditions: 1) when it reaches a point of material release, it jumps back to the next highest assembly and continues scheduling 2) when it reaches a drum, since these operations have already been scheduled, it returns to the nearest highest assembly, or 3) when it encounters an overload condition, it goes to the reminder list to insert the left-over load (the exact position depends upon the amount of left-over load) (Goldratt, 1990a:250). *DISASTER™* keeps all of this information in RAM. Since changes will almost certainly be made if a

new constraint is identified and the subordination process is repeated, no information is written to disk (Goldratt, 1990a:250). Only after *DISASTER™* is sure that all constraints have been identified, does it write the information to disk (Goldratt, 1990a:250).

Using the anticipated peak loads, *DISASTER™* is able to perform dynamic buffering to further reduce the size of the time buffers. The user provides the system with only the fixed portion of the buffer, based on pure Murphy. *DISASTER™* then adjusts the material release (the variable portion of the buffer) based on anticipated peak loads. Using the preceding procedure, the system then determines the impact of non-instant availability and adjusts the release date (the buffer) accordingly. The only figure that the user must supply is an estimate of "pure murphy' (Goldratt, 1990a:238). By adjusting the material release based on forecast of load, *DISASTER™* can significantly reduce inventory and lead-time (due to reduction in non-instant availability and therefore the time buffer) (Goldratt, 1990a:237).

After load leveling, a resource may be scheduled for processing at 100% capacity for several consecutive days (Goldratt, 1990a:239). Each day that a resource operates at 100% capacity, the length of its resource buffer is reduced by a percentage equal to the required protective capacity (Goldratt, 1990a:239). In TOC terminology, this reduction in the time buffer is referred to as a "hole" in the buffer-origin (Goldratt, 1990a:239). As the hole in the buffer-origin increases in size, the probability that Murphy will expose the constraint increases (Goldratt, 1990a:239). To ensure the constraint is not exposed, *DISASTER™* tracks the holes in each of the buffer-origins for resources with peak loads. Any hole larger than half of the time

buffer is then highlighted for corrective action (Goldratt, 1990a:240).

Peaks on red-lane activities need special attention since moving them backward may require modification of the constraint schedule itself (Goldratt, 1990a:241). If the date of the "peak" block is earlier than the order date minus the shipping buffer (that is, if sufficient slack exists), the peak load can be handled just like any other block (Goldratt, 1990a:241). In other cases, *DISASTER*™ has several options including the use of automatic overtime (if available) having the user authorize additional overtime, off-loading the block (the system identifies how much to off-load and when), or postponing the order (Goldratt, 1990a:242). If none of these options work, then the resource should be declared as another constraint and the system should try to identify and eliminate conflicts between the new and the old constraint (Goldratt, 1990a:242).

Before declaring a new constraint, *DISASTER*™ first looks for setup savings. If any savings are found on this resource, they will reduce the amount of required overtime and off-loading (Goldratt, 1990a:245). *DISASTER*™ also provides an option for treating glued blocks as one block. Since the blocks have already been glued (some blocks have already been repositioned with the resulting increased inventory), there is little to be lost, but potentially much to be gained (Goldratt, 1990a:245).

After the subordination stage is complete, the schedule will almost certainly contain multiple overload conditions on the first day (Goldratt, 1990a:252). *DISASTER*™ then categorizes the severity of the overloads by the amount of penetration into the particular resource's buffer, expressed as a multiple of the buffer length (i.e., two equals

two times the buffer length) (Goldratt, 1990a:253). The user should not rush to declare a constraint, since each additional constraint requires more protection (and thus inventory) (Goldratt, 1990a:253). On the other hand, when a resource is declared a constraint, more capacity on that resource is then freed and can be used to exploit the constraint (since constraints do not require protective capacity) (Goldratt, 1990a:253). When the user encounters the first-day peak problem, the major option is off-loading, so *DISASTER*™ also provides a list of off-loading opportunities (Goldratt, 1990a:254).

Subsequent Iterations. When *DISASTER*™ cycles through subsequent subordinations, unlike the first subordination run, it must check for conflicts between the old and the new constraints (Goldratt, 1990a:254). *DISASTER*™ uses the concept of rods to remove these conflicts in much the same manner as it did to remove conflicts between blocks of the same resource constraint during the first pass (Goldratt, 1990a:254). When constructing the ruins for the new constraint, *DISASTER*™ cannot place new blocks closer to half the buffer length from an old constraint block. If they are placed before the old constraint block, then they are called F(orward)-blocks since their rods need to be pointed forward in time to prevent them from being processed to close to the old block. Likewise, if they are place after the old block, then they are called B(ackward)-Blocks since their rods are pointing back in time (Goldratt, 1990a:255). Whenever a block to be processed on the new constraint is scheduled between two old blocks that already have rods (i.e., two processes that were done on the first constraint), these blocks are called B(ackward)F(orward)-Blocks since they require rods in both directions (Goldratt, 1990a:255).

DISASTER™ deals with ruins containing BF blocks differently than the standard ruins. First, it schedules the BF blocks. If a new block has to be scheduled in violation of a time rod, then the user must decide whether to modify the drum or off-load the block to another resource (Goldratt, 1990a:255). Once the BF blocks are scheduled, then *DISASTER™* considers them unmovable "rocks," and it then schedules the F blocks (Goldratt, 1990a:256). If scheduling the F-blocks necessitates that they be placed in the past, then the user again must decide whether to off-load or modify the drum (Goldratt, 1990a:256). If the drum has to be modified, then *DISASTER™* changes any old blocks that create violations to B-blocks, minimizing the amount of time they must be pushed into the future. From this point, *DISASTER™* then forgets the old constraint and repeats the process (Goldratt, 1990a:256).

Description of the Program

Overview. The *DISASTER™* scheduling software consists of three main modules: CALENDAR, NETGEN, and SCHEDULE. NETGEN and CALENDAR preprocess data for eventual use by SCHEDULE. The CALENDAR program allows the user to identify each resource's working hours for a specified period (the schedule horizon) (Avraham Y. Goldratt Institute, 1990a:1). NETGEN accepts data describing the production system in a general format, checks it for consistency, then repackages it into a concise form for scheduling (Avraham Y. Goldratt Institute, 1990c:1). Using information created from NETGEN and CALENDAR, the SCHEDULE module then produces schedules based on TOC principles: it maximizes the throughput of a plant by following an iterative process of identifying a resource constraint, exploiting it completely, and subordinating all other resources to meet the material needs of the identified

constraint(s) (Avraham Y. Goldratt Institute, 1990d:5). By acknowledging the existence of Murphy and statistical fluctuations, SCHEDULE generates an immune schedule (one that is relatively free from the effects of Murphy) that remains valid for a long time after release (Avraham Y. Goldratt Institute, 1990d:5).

Hardware requirements for *DISASTER*™ are minimal. Users need only an IBM AT or compatible microcomputer with at least an 80386SX processor. In addition the system requires a color monitor, at least 2 megabytes of extended memory, and at least two megabytes of hard disk storage. Additional on-line memory and storage space may be required, depending upon the specific application.

Conceptual Data Flow. The basic data required by the SCHEDULE block are product demands (what, how much, and when), and information about how the plant produces the product (usually found in the routings and bill of material (BOM) files) (Avraham Y. Goldratt Institute, 1990c:5). All of data required by *DISASTER*™ is already maintained in most plants; however, every plant has the data stored in various locations (Avraham Y. Goldratt Institute, 1990d:7). In addition, the data is usually in a format and structure that prohibits rapid and accurate processing (Avraham Y. Goldratt Institute, 1990c:1).

DISASTER™ simplifies the format problem by only specifying half of the required data interface--the remaining portion is specified by the particular user in accordance with any peculiar capabilities and requirements (Avraham Y. Goldratt Institute, 1990d:7). The user's half of the interface consists of preparing five ASCII text files, called the project data set (Avraham Y. Goldratt Institute, 1990d:8). *DISASTER*™ then accepts this information and creates a single file, the tasks

structure net, that is used as the primary input to SCHEDULE (Avraham Y. Goldratt Institute, 1990c:6). The problem for most companies is not the actual formatting and writing of the project data set files, but rather locating and collecting the required data from multiple sources. Usually, several databases are used, and often some required data is only available on paper (Avraham Y. Goldratt Institute, 1990c:18). Since much of the data will be dynamic in nature (i.e., customer orders), the user will need to develop automated procedures to periodically (i.e., daily) download the required files (Avraham Y. Goldratt Institute, 1990c:19).

Figure 6 summarizes the basic data flow requirements for *DISASTER™* (Avraham Y. Goldratt Institute, 1990c:21). As this figure clearly shows, it is not the intent of *DISASTER™* to replace any existing database(s); however, some of the database maintenance procedures may change (Avraham Y. Goldratt Institute, 1990c:21). For example, much of the data maintained under cost-driven systems will no longer be required (Avraham Y. Goldratt Institute, 1990c:21). Furthermore, data accuracy efforts need to concentrate primarily on data associated with the constraints (Avraham Y. Goldratt Institute, 1990c:22).

The Project Data Set. The project data set consists of five files: order, arrow, raw material, station, and resource (Avraham Y. Goldratt Institute, 1990b:29). Each of the project data set files is simply an ASCII list, with each line in the list referred to as a record and each separate piece of information within a record called a field (Avraham Y. Goldratt Institute, 1990c:25). The tasks structure net requires a certain set of data that represents a plant for scheduling,

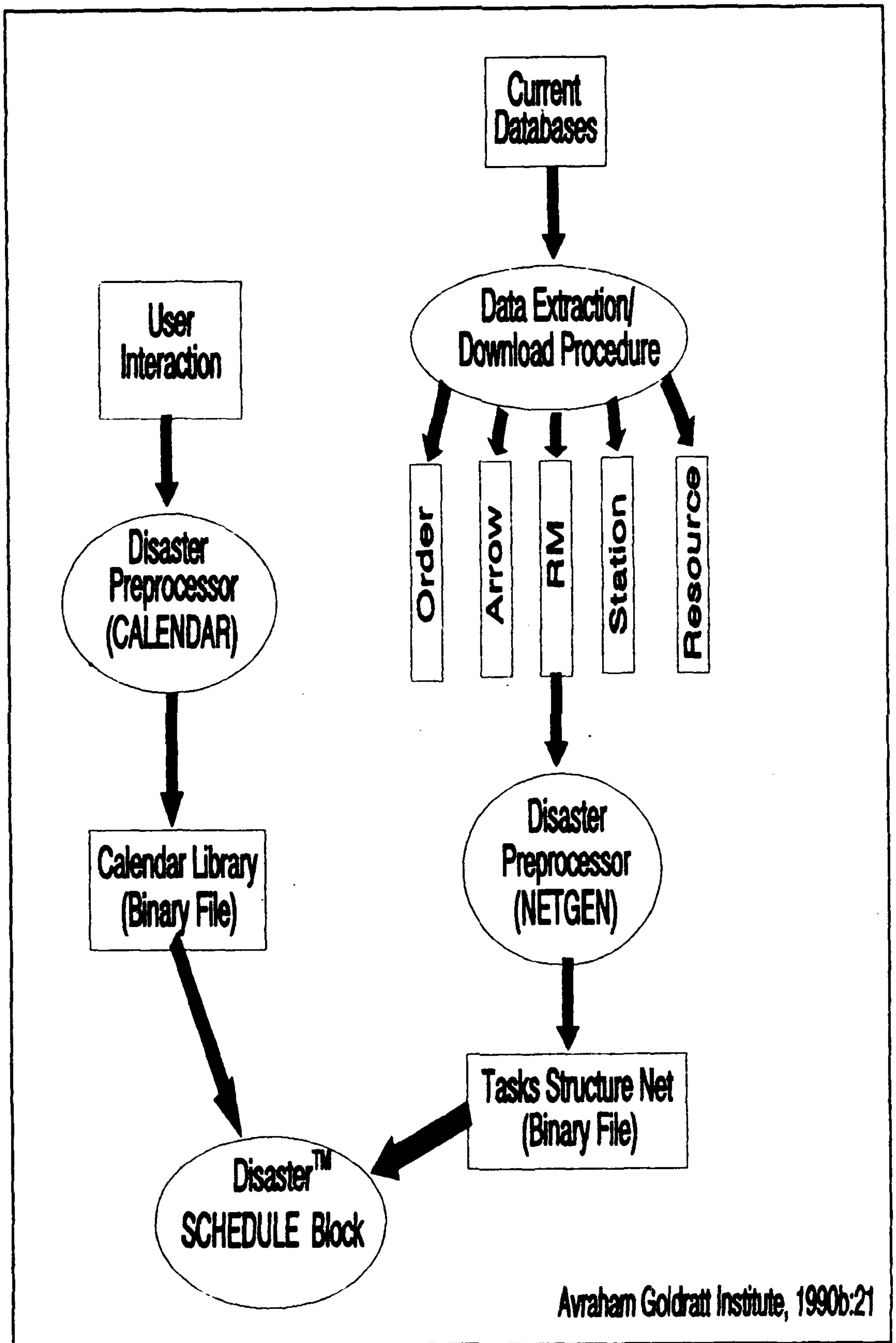

Figure 6. *DISASTER*™ Data Flow

called the *required* data. In addition, *optional* data that describes certain unique characteristics of the plant may also be specified. Both kinds of information are supplied in the project data set files (Avraham Y. Goldratt Institute, 1990c:25).

The documentation describes the content of the data files in depth; however, for purposes of this discussion, the following general descriptions are sufficient:

1. The *order file* contains a list of orders for all final products. It consists of a field for product name, quantity, and due date (Avraham Y. Goldratt Institute, 1990b:30).

2. The *arrow file* describes the flow of the raw materials (similar to a routings file), from the gating operations, through the processing stations, to the orders. This file contains the following fields: FROM name, the name of the first station; TO name, the name of the next station; quantity per, the amount of material that must be supplied to produce one unit of material; scrap or yield, the predictable scrap or loss rate between the two stations; and buffer size, an optional field identifying a buffer of specified size for placement at a particular arrow (Avraham Y. Goldratt Institute, 1990b:31).

3. The *raw materials file* consists of three required fields: raw material name; stock, the quantity currently in stock; and delivery time (in days). (Avraham Y. Goldratt Institute, 1990b:34).

4. The *station file* describes the process done at each stage of production on a specific part by a specific resource (Avraham Y. Goldratt Institute, 1990b:34). SCHEDULE must be able to distinguish

between an operation performed on a given part, and the same operation performed on a different part. Furthermore, it must also be able to distinguish between two operations performed on the same part with the same resource, but at different times (Avraham Y. Goldratt Institute, 1990c:6). To account for these distinctions, NETGEN requires that each part/operation combination be uniquely identified (Avraham Y. Goldratt Institute, 1990c:6). Each station that is referred to in the arrow file must be included in the station file (Avraham Y. Goldratt Institute, 1990b:34).

The station file contains the following required fields: station name, a unique identifier for the part number and the operations to be performed; resource name, of the resource used at the station; processing time, the time in minutes to process a single part or unit; work in process, the quantity of items completed at this station and currently available; and setup time, the time in minutes to perform the operation at this station (Avraham Y. Goldratt Institute, 1990b:34-36). The station file also contains the optional field time per batch to distinguish between per unit and per batch processing time specifications (Avraham Y. Goldratt Institute, 1990b:36).

5. The *resource file* describes "every means of production available to the plant," including both workers and machines (Avraham Y. Goldratt Institute, 1990b:37). This file contains two required fields, resource name and resource quantity, and two optional fields, calendar and protective capacity (Avraham Y. Goldratt Institute, 1990b:37). The calendar field allows the user to specify whether a resource works according to a resource-specific calendar (versus the default calendar). This field is necessary in cases such as when one department works

different hours than the rest of the company (Avraham Y. Goldratt Institute, 1990b:37). The protective capacity field allows the user to specify more or less protective capacity than provided under the default parameter screen (described below) (Avraham Y. Goldratt Institute, 1990b:38).

DISASTER™ recommends that the five project data files be created in the order presented above (Avraham Y. Goldratt Institute, 1990c:27). Beginning with generation of the order file will naturally lead to each of the other files (Avraham Y. Goldratt Institute, 1990c:27). The user does not need to be overly concerned with data accuracy during the project data set development. The only important data is the data concerning the constraint, and each time the user has to make a decision with this data during scheduling, the SCHEDULE module presents it for verification (Avraham Y. Goldratt Institute, 1990c:18).

Running NETGEN. The NETGEN module accepts the five project data set files and "packages" the information into one standard file that is used as the primary input to SCHEDULE (Avraham Y. Goldratt Institute, 1990b:29). NETGEN requires that each of the files be sorted prior to creating the tasks structure net. Depending upon the user's capability, it is often preferable to complete this sorting prior to downloading; however, if the project data set files have not been previously sorted, NETGEN will also sort and re-write them. Sorting is important to speed up any subsequent processing on these files and to assist in identifying and highlighting errors (Avraham Y. Goldratt Institute, 1990b:40).

During processing by NETGEN, each of the required project data set files must be in one directory (Avraham Y. Goldratt Institute, 1990c:45). NETGEN first performs a variety of tests to ensure the data

is accurate and consistent. If any errors are located, NETGEN will output a separate error file instead of the tasks structure net. This file contains a specific error message for each error encountered during execution (Avraham Y. Goldratt Institute, 1990b:40). Once the project data set passes all the screening tests, NETGEN produces the tasks structure net. This net is basically a map of how material flows through the plant from raw materials to finished goods (Avraham Y. Goldratt Institute, 1990d:7).

CALENDAR. In addition to the tasks structure net, the other main input to the SCHEDULE program is the calendar file. The calendar library file, a binary file created by the CALENDAR program, identifies each resource's working hours over the schedule horizon (Avraham Y. Goldratt Institute, 1990a:1). CALENDAR produces two types of calendars: a default calendar identifies working hours common to the majority of plant resources, and a resource-specific calendar which identifies working hours unique to specific resources (Avraham Y. Goldratt Institute, 1990a:1). When creating the default calendar, CALENDAR asks for five parameters: start date, end date, default work hours, work hours, and calendar name (Avraham Y. Goldratt Institute, 1990a:11). After these values are supplied, CALENDAR then fills in the specified working hours for each day in the specified range (the schedule horizon). The user can then go back and edit the days for special work hour requirements (i.e., weekends or other unique processing requirements) (Avraham Y. Goldratt Institute, 1990b:43). If some resources do not work according to the default calendar hours, the user can then create separate resource-specific calendars (Avraham Y. Goldratt Institute, 1990b:44). All of these calendars are created from

the CALENDAR program's main menu. When the user is satisfied that all working hours have been appropriately specified, the files are saved in the calendar library file (Avraham Y. Goldratt Institute, 1990b:44).

SCHEDULE.

General. Once the tasks structure net and the calendar library are supplied to SCHEDULE, they are placed in the computer's memory and reside there during schedule generation (Avraham Y. Goldratt Institute, 1990d:8). The SCHEDULE program takes the tasks structure net, the calendar library file, and a parameter file and schedules the plant's resources while accounting for open customer orders, the time window the user wishes to schedule the resources (the schedule horizon), internal constraints, and non-constraint resources (Avraham Y. Goldratt Institute, 1990b:47). SCHEDULE maximizes the throughput of a plant by following an iterative process of identifying a constraint, exploiting it completely, and subordinating all other resources to meet the material needs of the identified constraints (Avraham Y. Goldratt Institute, 1990d:5). During the subordination step, conflicts between the resources are revealed and new constraints emerge, marking the beginning of a new iteration (Avraham Y. Goldratt Institute, 1990d:5). Iterations continue until all conflicts have been identified and resolved (Avraham Y. Goldratt Institute, 1990d:5). Since only a very small number of constraints can exist in any viable system, few iterations will be necessary (Avraham Y. Goldratt Institute, 1990d:5).

The basic philosophy of *DISASTER*™ is that it should be a "white box"--nothing is hidden from the user. If any data processing is done, the user should "have the capability to grasp at least a conceptual meaning of the processing" (Avraham Y. Goldratt Institute, 1990b:13).

In addition, whenever data is derived as a result of such processing, then the user should be given access to it (Avraham Y. Goldratt Institute, 1990b:14-5). Another objective of the user interface is that it should provide maximum flexibility--it should not constrain the user from taking any action unless there are good reasons for doing so (Avraham Y. Goldratt Institute, 1990b:15). The user interface seeks to focus the user's attention on the important issues while providing freedom to seek as much detail as desired (Avraham Y. Goldratt Institute, 1990b:15).

The SCHEDULE program is entirely menu-driven (Avraham Y. Goldratt Institute, 1990b:51). The structure of the *DISASTER™* user interface is a series of screens, each associated with one or more processing steps (Avraham Y. Goldratt Institute, 1990b:15). The hierarchy enables the user to obtain additional detail concerning each processing stage by stepping down through a series of related screens. In this manner, the user controls how much detail is provided (Avraham Y. Goldratt Institute, 1990b:15).

Running SCHEDULE. Execution of the SCHEDULE module results in six main screens, one for each of the six steps *DISASTER™* uses to schedule: parameters, identification, ruins, drum, late order list, and subordination (Avraham Y. Goldratt Institute, 1990b:52). For each screen, SCHEDULE provides *explore* options that enable the user to view information in more detail (Avraham Y. Goldratt Institute, 1990b:52).

Parameter Screen. SCHEDULE uses ten parameters that inform the program of the "boundaries" within which it is to operate. These parameters can be classified into three categories: horizon (date range), buffer sizes, and system defaults (Avraham Y. Goldratt

Institute, 1990b:58). The horizon provides the start and end dates for the scheduling horizon (Avraham Y. Goldratt Institute, 1990b:59). Three types of buffers are used by SCHEDULE: resource, shipping, and assembly buffers (Avraham Y. Goldratt Institute, 1990b:59). The sizes for these buffers are always provided in hours, and they are determined by a variety of factors (see buffer sizing discussion above) (Avraham Y. Goldratt Institute, 1990b:59). As discussed previously, the shipping buffer influences the *effective* horizon since, for determination of capacity, jobs with due dates before the horizon plus the shipping buffer are always considered (Avraham Y. Goldratt Institute, 1990b:59).

The system defaults include data for work hours per day, automatic daily, weekly, and weekend overtime, and protective capacity. These parameters are only used if no information for them was provided in the tasks structure net input data (Avraham Y. Goldratt Institute, 1990b:60). The overtime parameters all refer to overtime the user wants the system to allocate automatically when peak requirements are present in the schedule. In these cases, SCHEDULE assigns overtime according to specified limits prior to checking with the user about assigning additional overtime use (Avraham Y. Goldratt Institute, 1990b:60). The last parameter, protective capacity, permits the user to specify a default amount of protective capacity for resources that do not have another amount entered via the resource file (Avraham Y. Goldratt Institute, 1990b:61). The protective capacity is entered as a percentage of the total capacity, then *DISASTER*™ avoids using any of this capacity during scheduling (Avraham Y. Goldratt Institute, 1990b:61).

Identification Screen. The identification screen contains two kinds of information: a list with all of the resources, sorted by the ratio of load to capacity, and a station load histogram accompanying the resource list (Avraham Y. Goldratt Institute, 1990b:63). The resource list is a list of all resources, sorted by the ratio of load to capacity, that contains information such as the number of resource units, the load without setup, the amount of load available, and the total percent load (Avraham Y. Goldratt Institute, 1990b:63). The load histogram provides the user with a graphical presentation of the loads placed on each resource (Avraham Y. Goldratt Institute, 1990b:64).

Explore options are available for both the resource list and the histogram (Avraham Y. Goldratt Institute, 1990b:64). The resource list explore option allows the user to "jump" into the list and scroll through the various resources. As the user does so, the histogram relevant to each resource is presented (Avraham Y. Goldratt Institute, 1990b:64). The histogram explore option enables the user to obtain the underlying information relating to each highlighted resource (Avraham Y. Goldratt Institute, 1990b:64).

The primary purpose of the identification screen is to verify the data. As noted earlier, not all the data is important, so SCHEDULE presents only the data used to determine that a particular resource is a constraint. *DISASTER*™ provides all necessary data to verify that a resource is indeed a constraint, then the user must verify that the data is correct (Avraham Y. Goldratt Institute, 1990b:67). Once the user declares that the data is valid, then the system will declare the resource as a constraint (Avraham Y. Goldratt Institute, 1990b:69).

Even though the system can easily determine which resources are loaded greater than 100%, the user always makes the choice as to which constraint should be used to develop the schedule (Avraham Y. Goldratt Institute, 1990b:69). If the user selects the market as the constraint, then the system will move directly to subordination (the ruins, drum, and late order list screens are not relevant); however, if the user selects a resource as the constraint, SCHEDULE then moves to the ruins screen (Avraham Y. Goldratt Institute, 1990b:71).

Ruins Screen. The ruins screen shows the *ideal* processing time of each batch or "block" of work on the resource constraint. If the block is performed earlier (farther to the left), then the material will be available earlier than needed, resulting in excess work-in-process inventory. On the other hand, if the block is processed later (farther to the right), then it will eat into its buffer time, increasing the probability that Murphy would delay it (Avraham Y. Goldratt Institute, 1990b:71). The ruins screen's axis is the schedule horizon (Avraham Y. Goldratt Institute, 1990b:71). This screen presents the blocks at their ideal processing time, assuming no capacity limitations. In reality, if the resource is a constraint, there will be more required blocks to process than units available on the resource (Avraham Y. Goldratt Institute, 1990b:72).

Using the explore option for the ruins screen, the user is permitted to move the cursor to specific blocks and obtain information such as the block's ideal start and end dates, daily load data, and rods (Avraham Y. Goldratt Institute, 1990b: 75-6). When the daily load calculation data is requested, *DISASTER*™ presents a graphical display of the amount of load relative to available capacity for each day of the

schedule horizon (Avraham Y. Goldratt Institute, 1990b:79). When information about rods is requested, SCHEDULE presents a list of all rods that are connected to the active block (the block the cursor is highlighting) (Avraham Y. Goldratt Institute, 1990b:77). Once the user is comfortable with these data, the next step is to begin leveling the load: the drum screen.

Drum Screen. The drum screen displays the constraint resource's batches after they have been leveled, so the blocks are no longer placed based on ideal processing times (Avraham Y. Goldratt Institute, 1990b:79). The drum screen reflects the actual times that the blocks must be performed to satisfy the finite availability of the resource and all prior throughput decisions (i.e., any previous drums and any orders that fall within the scheduling horizon) (Avraham Y. Goldratt Institute, 1990b:79). Any batch that is now scheduled to be completed later than one half of the shipping buffer after its ideal processing time is colored red: these blocks will almost certainly be late (Avraham Y. Goldratt Institute, 1990b:81).

Using the drum screen explore options, the user can allocate overtime and identify setups. At any time, the user can undo any previous actions or stop and request a presentation of summary statistics (Avraham Y. Goldratt Institute, 1990b:81). For setups, the user can simply instruct *DISASTER*™ to perform setup savings, and the system will glue all relevant orders (given the move will not affect the completion time of other jobs). In addition, the user can specify a setup ratio that limits the amount of time that a job can be "backed up" for gluing (Avraham Y. Goldratt Institute, 1990b:85). When the user selects the overtime option, *DISASTER*™ assigns overtime based on

specified limits. These limits can either be set for all resources, using the parameter screen, or they can be set for specific resources using a utility (Avraham Y. Goldratt Institute, 1990b:85).

Some orders may still be late after global overtime and setup savings have been identified, so *DISASTER*™ also provides "batch manipulation" options that can be applied to specific blocks (Avraham Y. Goldratt Institute, 1990b:85). The primary batch manipulation options are assignment of additional overtime, offloading, splitting, or simply rescheduling a particular batch (Avraham Y. Goldratt Institute, 1990b:89-90). Once the user is satisfied that all options have been exhausted, he or she then needs to see the results of all decisions, especially with respect to late orders.

Late Order List Screen. The purpose of the late order list screen is to show the impact of drum decisions on specific orders by identifying which orders were made late and by how long (Avraham Y. Goldratt Institute, 1990b:91). This screen enables the user to search for specific orders, to list *all* orders (only now are the late ones displayed in red), or to move the cursor through the list of late orders to obtain more specific information (Avraham Y. Goldratt Institute, 1990b:93). After this step, SCHEDULE is ready to begin subordination.

Subordination Screens. Each time a constraint has been exploited, *DISASTER*™ must schedule the non-constraints to support these (as well as prior) throughput decisions. This process is called subordination (Avraham Y. Goldratt Institute, 1990b:93). There are two primary screens presented during the subordination process: the red lane peak (RLP) screen and the first day load (FDL) screen (Avraham Y. Goldratt Institute, 1990b:95). The RLP screen may occur multiple times,

corresponding to any time a red-lane peak occurs, but the FDL screen occurs only when *DISASTER*™ reaches the present, at the end of its processing (Avraham Y. Goldratt Institute, 1990b:95).

RLP Screen. As discussed previously, *DISASTER*™ performs backward scheduling on non-constraints, beginning with the latest date of the effective schedule horizon and scheduling all loads for each day before moving to the next earlier day (Avraham Y. Goldratt Institute, 1990b:95). TOC refers to this consistently-backward movement as the "uniform time front" (Avraham Y. Goldratt Institute, 1990b:95). During subordination, *DISASTER*™ identifies peaks in demand and pushes these peaks earlier in time (Avraham Y. Goldratt Institute, 1990b:95). A red lane is the portion of a net that is fed by a constraint resource (Avraham Y. Goldratt Institute, 1990b:95). For a peak on a red lane activity, *DISASTER*™ can push the peak load earlier only if it is not pushed before the scheduled start time of the feeding constraint batch (Avraham Y. Goldratt Institute, 1990b:95).

DISASTER™ refers to peaks in demand that cannot be pushed earlier as red lane peaks (RLP) (Avraham Y. Goldratt Institute, 1990b:95). If left unresolved, these peaks will usually cause a hole in region one of the buffer that the non-constraint is feeding, so the user must attempt to resolve this peak before subordination can continue (Avraham Y. Goldratt Institute, 1990b:95). Before the user attempts to resolve the peak, *DISASTER*™ first displays all data relevant to the overload condition for verification.

Once the data have been verified, the user next attempts to resolve the red lane peak. The RLP screen provides six options relevant to resolving these peaks: overtime, offload, push order due-date,

ignore, next drum, and halt (Avraham Y. Goldratt Institute, 1990b:97). Selection of the overtime option results in a display identifying the amount of overtime required on the resource and the window of time (defined by the peak date and the peak date plus buffer) within which the overtime should be used to resolve the peak (Avraham Y. Goldratt Institute, 1990b:97). The offload option identifies how many pieces need to be offloaded, then the user must specify the desired number of pieces to offload and the receiving resource (Avraham Y. Goldratt Institute, 1990b:97). Some situations may arise when the user wants to ignore the peak (i.e., there may be some other temporary fix that enables the resource to complete the load). When the ignore option is selected, *DISASTER™* does not disregard the peak load, but rather treats it as though it is not a problem and continues scheduling (Avraham Y. Goldratt Institute, 1990b:98). The push order due date option is displayed only when the RLP is directly feeding a shipping buffer. In these situations, the due date will be pushed back by a time equal to the size of the peak, thus providing additional time for the resource to complete processing (Avraham Y. Goldratt Institute, 1990b:98). The next drum option allows the user to specify the resource experiencing the peak in demand as a secondary resource constraint. *DISASTER™* then proceeds immediately to create the ruins for this resource (Avraham Y. Goldratt Institute, 1990b:98). The halt option stops subordination and returns to the main menu (Avraham Y. Goldratt Institute, 1990b:98).

FDL Screen. As subordination continues, SCHEDULE will push any peak in demand for a non-constraint resource earlier in time (Avraham Y. Goldratt Institute, 1990b:98). Since it is impossible to push peak demands earlier than the start time of the

effective scheduling horizon, *DISASTER*™ places these demands on the first day (Avraham Y. Goldratt Institute, 1990b:98). As a result, resources will likely experience first-day demands in excess of available capacity. *DISASTER*™ calls these loads first day load (FDL) peaks (Avraham Y. Goldratt Institute, 1990b:98). FDL peaks indicate that subordination has been unsuccessful at subordinating the non-constraints to the existing drum(s) and revised order due dates (Avraham Y. Goldratt Institute, 1990b:98-9). Schedules containing these peaks would be unrealistic, so *DISASTER*™ identifies each FDL situation and provides explore options for resolving them (Avraham Y. Goldratt Institute, 1990b:99).

There are six explore options available under the FDL screen: peak list, batches, next drum, ignore, overtime, and halt (Avraham Y. Goldratt Institute, 1990b:101). The peak list option permits the user to scroll through the resources listed on the peak list and obtain additional information and displays (Avraham Y. Goldratt Institute, 1990b:101). The batches option allows the user to scroll through the list of batches that comprise a particular FDL. *DISASTER*™ then displays additional information about the batch and permits the user to either offload or assign additional overtime to try to resolve the FDL (Avraham Y. Goldratt Institute, 1990b:102-3). The next drum, ignore, and halt options are analogous to those options available under the RLP screen (Avraham Y. Goldratt Institute, 1990b:103-4). The overtime option assigns overtime to resolve the entire FDL peak of the resource (in contrast to the batches overtime option which only concentrates on a particular batch) (Avraham Y. Goldratt Institute, 1990b:104).

Whenever the user opts to select a secondary constraint (the next drum option), *DISASTER™* proceeds immediately to the ruins screen for the new constraint (Avraham Y. Goldratt Institute, 1990b:104). The ruins for secondary constraints differs slightly from the ruins for the first drum. For secondary constraints, *DISASTER™* must consider interaction between this new constraint and all previous resource constraints (Avraham Y. Goldratt Institute, 1990b:104). SCHEDULE handles this interaction by placing buffers between batches on one constraint and batches on another constraint, ensuring at least one half a resource buffer is always between these two blocks (Avraham Y. Goldratt Institute, 1990b:104). Buffers are maintained by using the concept of rods, and information about these rods can be accessed through the ruins screen (Avraham Y. Goldratt Institute, 1990b:104). After the ruins are created, *DISASTER™* then establishes a drum for the secondary constraint while considering all prior drums fixed (Avraham Y. Goldratt Institute, 1990b:104). Rods are used to synchronize the activities of the current drum with the activities of previous drums while ensuring adequate buffers are maintained (Avraham Y. Goldratt Institute, 1990b:105).

Fix Drum Violations Screen. In some cases, all batches of the secondary constraint might be placed on the drum with no violations; however, often the rods prohibit placement of the new batches on the drum. In these cases, the user must fix these violations before proceeding, so SCHEDULE presents a modified drum screen, called the fix drum violations screen (Avraham Y. Goldratt Institute, 1990b:105). This screen provides four options for fixing violations: offloading, overtime, shrink rods, and drum loop (Avraham Y. Goldratt

Institute, 1990b:105). When the offload option is selected, *DISASTER™* requests the name, the setup time, and the process time for the alternate resource (Avraham Y. Goldratt Institute, 1990b:107). The overtime option is only available if *DISASTER™* determines that overtime will help solve the violation. If advantageous, *DISASTER™* assigns overtime in the amount necessary to solve the violation (Avraham Y. Goldratt Institute, 1990b:107). The shrink rods option allow the user to change the length of the rods, enabling the batches to fit on the schedule. Using this option reduces the buffers; therefore, it also reduces the level of protection (Avraham Y. Goldratt Institute, 1990b:107). The drum loop option enables the user to return to a previous constraint and modify that resource's rods to remove the violations; however, when this is done, the user loses much of the processing that was done prior to this resource's ruins screen (Avraham Y. Goldratt Institute, 1990b:109).

SCHEDULE Outputs. After the user has looped through identification, exploitation, and subordination for all the system's constraints, *DISASTER™* presents a message stating that no more resources exist with first day peaks and asks if the user wants schedules to be written (Avraham Y. Goldratt Institute, 1990b:111). Output from SCHEDULE consists of eleven files, six pertaining specifically to the scheduling of resources and five information files (Avraham Y. Goldratt Institute, 1990b:113).

The Constraint File. The constraint file contains the schedules for the stations of each constraint resource. This file consists of a number of records, each describing the processing of a batch on that particular resource at a specific point in time (Avraham

Y. Goldratt Institute, 1990b:115). Each record in the constraint file includes a field identifying the batch number, the constraint resource name, the station's name, the order for which the batch is required, the number of pieces in the batch, the processing time, the setup time, the ideal end date and time, the actual date and time processing should begin, expected completion date and time, the particular unit on which the batch should be processed, and identification of whether the processing time is per part or per batch, the number of the drum that this batch feeds (if it has an F-rod), and the number of the drum to which it is attached (if the batch has B-rods).

The Non-Constraint File. The non-constraint file contains schedules for the non-constraint resources. Like the constraint file, this file contains a number of records that describe specific batches to be processed, but on the non-constraint resource (Avraham Y. Goldratt Institute, 1990b:116). Non-constraint schedules are not required for most non-constraint resources. Instead, operators are instructed to simply process material as soon as it arrives (Avraham Y. Goldratt Institute, 1990b:116). The non-constraint schedule contains the following fields: resource name, station name, amount of material to be processed, date before which processing should *not* start, setup time for the station, processing time for the station, identification of whether processing time is per part or per batch (Avraham Y. Goldratt Institute, 1990b:116).

The New Order Due-Date File. The new order due dates file provides the status of each order that was already within the schedule horizon when schedule began (Avraham Y. Goldratt Institute, 1990b:117). It is likely that some of these orders will have revised due dates and

some may even have been pushed out of the effective schedule horizon (Avraham Y. Goldratt Institute, 1990b:117). This file contains the following fields: order name, new order due date, order quantity, and whether the order is on-time or late, whether the order is inside or outside the effective horizon (Avraham Y. Goldratt Institute, 1990b:117).

The Pick List File. The pick list file provides a schedule of raw material release to specific gating operations (Avraham Y. Goldratt Institute, 1990b:117). This file includes the following fields: resource name, station name for the gating operation, type of raw material, amount of raw material, and required release date for the raw material (Avraham Y. Goldratt Institute, 1990b:117).

The Overtime File. The overtime file lists the amount and the time of assignment for each resource that received overtime (Avraham Y. Goldratt Institute, 1990b:117). This file consists of three fields: resource name, date of assignment, and number of hours assigned (Avraham Y. Goldratt Institute, 1990b:117).

The Raw Material File. The raw material file contains two sections. The first section, intended to identify how much raw material is in stock, includes two fields: raw material name and amount in stock (Avraham Y. Goldratt Institute, 1990b:117). The second portion of this file is intended to identify when and how much of each raw material is needed to satisfy the constraints. This section includes the following fields: raw material name, net amount needed to be ordered, date material is needed, and material delivery lead time (Avraham Y. Goldratt Institute, 1990b:118).

The Information Files. In addition to the files necessary for scheduling the plant, SCHEDULE also maintains additional files containing information that may be accessed by the user. These files include a resource file, a modification log, a program activity log, a screen dump file, and a parameters file. In addition, *DISASTER™* maintains files for current and previous keystrokes that SCHEDULE may use during future processing (Avraham Y. Goldratt Institute, 1990b:115). The content of each of the information files is discussed in depth in the *DISASTER™* documentation.

Anticipated Benefits of DISASTER™. While the DBR concept may appear similar to MRP and MRP II scheduling, MRP never successfully integrated the master production schedule into the scheduling process. Umble and Srikanth propose three major reasons for this shortcoming: 1) MRP provides no systematic procedure for developing a *valid* MPS; 2) MRP is geared to producing local optimums rather than optimizing the system as a whole; and 3) MRP fails to recognize the conflict that exists between the requirements for success of the production system and prevailing management policies and performance evaluation procedures (Umble and Srikanth, 1990:138).

DBR differs significantly from previous scheduling systems because it 1) analyzes requirements to achieve a smooth production flow throughout the entire plant (it supports global vice local objectives), 2) explicitly recognizes and resolves system conflicts, and 3) provides a systematic procedure for managing problems resulting from Murphy and inaccurate data (Umble and Srikanth, 1990:139).

One of the major advantages of *DISASTER™* is its recognition of the level of disturbances on the shop floor. The designers of

DISASTER™ have clearly stressed the importance of considering the effect of disturbances during the scheduling process. By using appropriate levels of protective inventory and protective capacity, *DISASTER*™ may significantly improve the ability of companies to produce realistic schedules. In addition, the system also provides simplicity, faster processing times, and reduced data requirements and accuracy.

Simplicity. One of the most noted difference between the DBR scheduling technique and traditional methods such as MRP and MRP II is the simplicity of the TOC method: instead of trying to schedule and prioritize everything, TOC simply limits the release of material to the floor and permits non-constraint operations to work on jobs as they arrive (Umble and Srikanth, 1990:164). Unlike traditional scheduling, only a very limited number of schedules (corresponding to schedule release points) must be developed when one uses the DBR technique, and even these schedules are based on the constraint schedule (Trigger:79). Furthermore, a key consideration in DBR is that only these schedule release points must be strictly controlled: other points require only minimal control since they are activated by the arrival of material (Umble and Srikanth, 1990:167). This simplification of the scheduling process is a major advantage of *DISASTER*™.

Processing Speed. Schedule generation in MRP systems is often time consuming, ranging from several hours for a small plant to an entire weekend for a more complex organization (Goldratt, 1990a:164). The schedule generation time is important to *DISASTER*™ because the schedule block is not an end in itself--it is the basis for both the control and what-if modules. Without it, managers cannot make good decisions (Goldratt, 1990a:164).

The fast processing time of SCHEDULE is a result of the fact that NETGEN and CALENDAR put required data into a concise form that fits directly into the computer's memory (Avraham Y. Goldratt Institute, 1990d:7). In addition, the architecture of the tasks structure net also eliminates the need to continually switch between BOM and routing files (Avraham Goldratt Institute, 1990d:7). NETGEN also combines both stores and WIP inventory into one file, and codes both according to the last process performed (Goldratt, 1990a:175). Using today's computers, with tremendous amounts of on-line memory, *DISASTER™* accesses the disk one time, then holds everything in RAM while performing the calculations. In this manner, schedules for large systems can now be produced in less than one hour (Goldratt, 1990a:176).

Data Requirements and Accuracy. Since information is built in a hierarchical structure and the decision process allows the system to go from one level to another, changing to a new decision process changes the nature of the required data (Goldratt, 1990a:83). In addition, a new decision process changes the required data accuracy (Goldratt, 1990a:85). Not only do current MRP approaches require an enormous amount of data, they fail to identify what specific data needs to be verified. Instead, they only declare that the data must be "95 percent accurate," and leave the rest to the user (Goldratt, 1990a:187).

The data accuracy and availability problem is significantly reduced with *DISASTER™*. Product costs (as traditionally measured) are no longer important. Under the throughput world, only the data associated with the constraint is important. Instead of attempting to ensure that *all* data is complete and accurate, managers need only concentrate on data associated with the constraint (Goldratt, 1990a:84).

For example, with the gedunken experiment of Figure 4, the processing times of the non-constraint resources could vary significantly without affecting the outcome of the experiment. *DISASTER™* greatly simplifies the data availability problem by recognizing that only a few things are important. Unlike MRP, *DISASTER™* clearly identifies what data needs to be verified.

Chapter Summary

Up to this point, the research has examined traditional manufacturing approaches (MRP and MRP II), the application of MRP II to AFLC's DMMIS, JIT manufacturing, and TOC. This chapter has examined the the *DISASTER™* system in depth. First, the conceptual basis for its operation was discussed, including the program's decision process, the criteria for a good schedule, and the logic of its scheduling procedure. Next, the characteristics and operation of its major software modules, CALENDAR, NETGEN, and SCHEDULE, were examined. The required data flow (development of the project data set by NETGEN and the calendars by CALENDAR) was identified, the major steps used by the SCHEDULE module during scheduling were discussed, and the primary system outputs were identified. Now that the characteristics of *DISASTER™* have been outlined, the research will now focus on its application within a commercial manufacturing company.

V. Case Analysis: The Zycon Corporation

Scope

This case analysis is analytic in nature, intended to document the use of *DISASTER™* in a real-world environment. The primary purpose of the analysis is not to provide a complete review of TOC implementation at Zycon; however, since the success of *DISASTER™* is highly dependent upon the company's understanding and use of TOC principles, relevant aspects of TOC are also examined. This chapter begins by exploring characteristics of Zycon's product, printed circuit boards (PCBs), and the PCB industry. Next, this section provides an overview of Zycon's background, including their marketing strategy, status within the industry, production process, and TOC implementation background and status. Finally, this chapter analyzes the implementation of *DISASTER™* at Zycon.

Industry Background

The Product: Printed Circuit Boards (PCBs). A PCB is composed of a dielectric material to which metallic patterns are bonded (Shoemaker, 1991a). Printed circuit boards (PCB) are interconnecting components in electronic devices such as microprocessors, semiconductors, and other integrated circuits (Shoemaker, 1991a). When assembled, these boards are the basic components for nearly all electronic systems (Shoemaker, 1991a). Boards are commonly classified with respect to rigidity (either rigid, flexible, or rigid-flex) and number of layers (one-sided, two-sided, or multilayer) (Shoemaker, 1991a). Multilayer boards are composed of several layers of circuitry, laminated together to form a single board (Shoemaker, 1991a). As shown in Figure 7, the major end

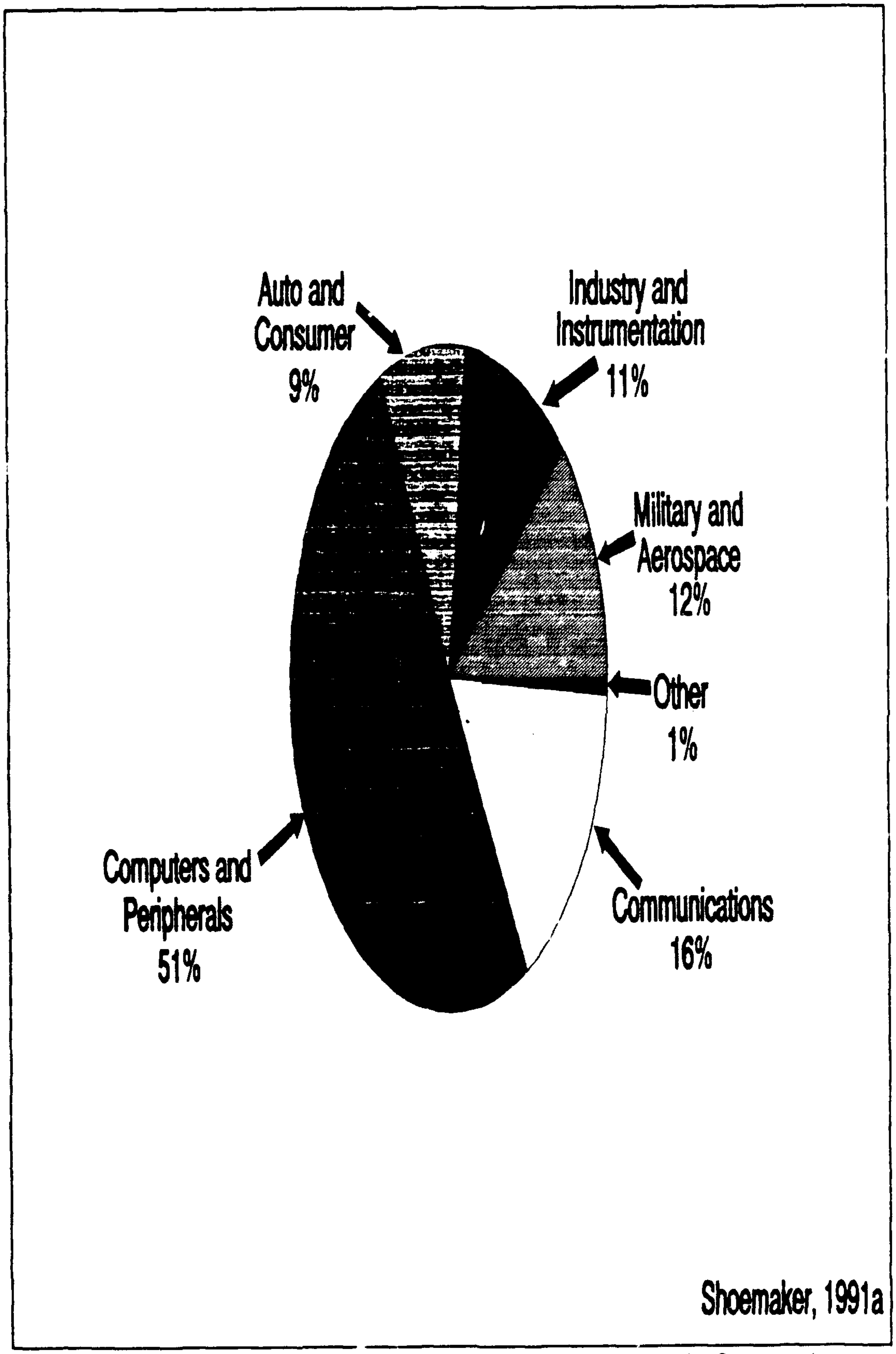

Figure 7. PCB Applications: Major Market Segments

uses of PCBs are computer/computer-related, communications, and industrial/instrumentation applications (Shoemaker, 1991a).

The Market. There are approximately 600 domestic board manufacturers (S oemaker, 1991a). Over the past seven years, the domestic market has grown at an average annual rate slightly greater than 10 percent (Shoemaker, 1991a). As shown in Figure 8, the industry experienced a rapid growth rate from 1983 to the beginning of 1985 (Shoemaker, 1990). In 1985, however, the industry declined sharply. During 1986, the growth rate of U.S. PCB shipments declined by approximately 27 percent (Shoemaker, 1990). This market downturn resulted in the consolidation of over 100 manufacturers. The market rebounded in 1987, but between 1987 and 1989 another 200 firms consolidated (Shoemaker, 1991a).

The current industry trend is towards smaller, higher performing, sophisticated systems. To meet this demand, more and more companies are now producing multilayer PCBs (see Figure 9). Leading computer-related product manufacturers have traditionally competed on the basis of quality and time to market rather than price (Shoemaker, 1991a).

Introduction to Zycon

Background. The Zycon Corporation is a privately held manufacturer of circuit boards that was co-founded by its President and three of its present senior vice presidents in 1976. Zycon is recognized as one of the PCB industry's technological leaders. Multilayer jobs comprise a significant percentage of the company's work (Shoemaker, 1991a). Zycon is currently the fourth largest producer of printed circuit boards among independent manufacturers (Shoemaker, 1991a).

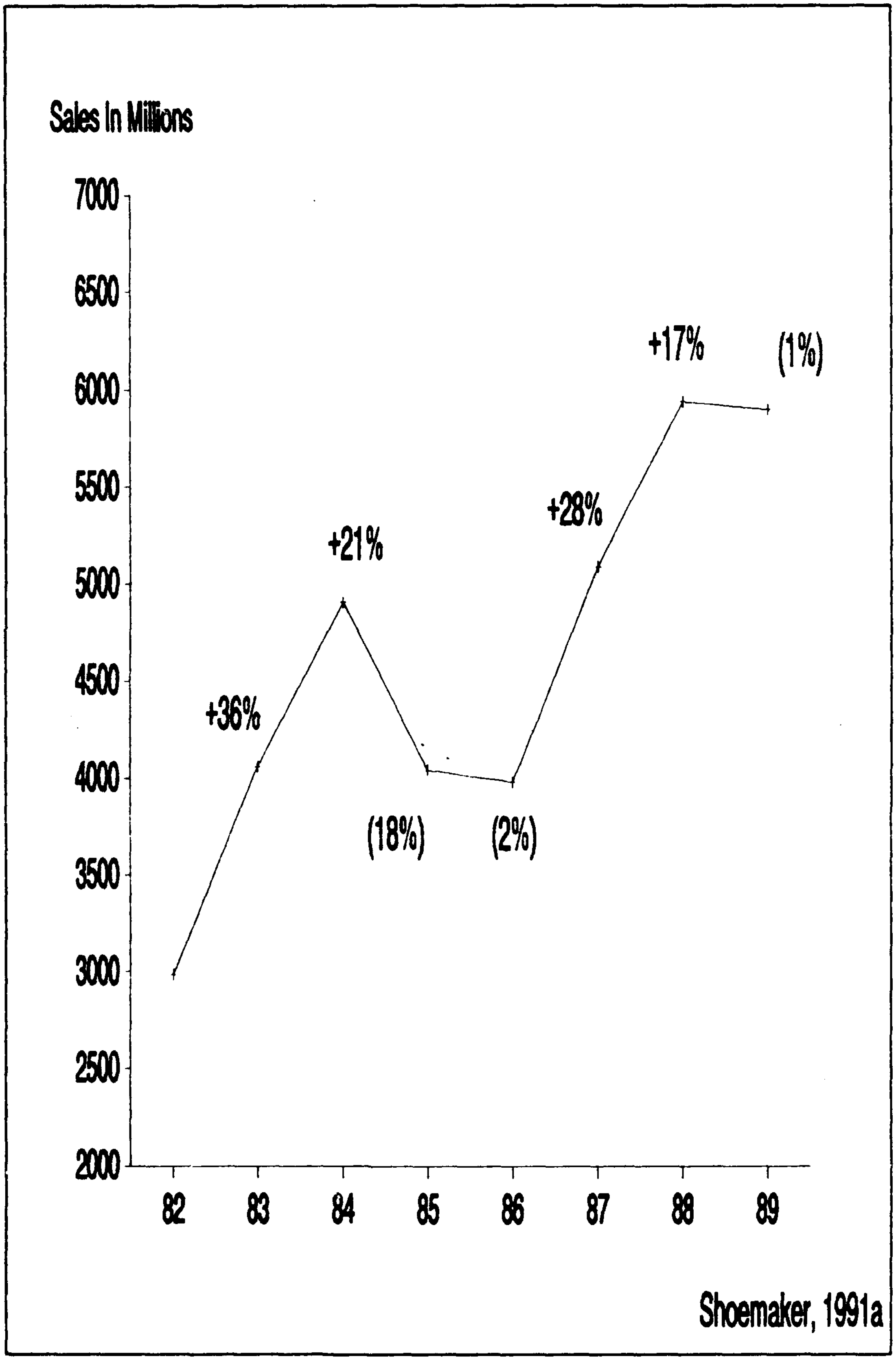

Figure 8. Domestic PCB Industry Growth

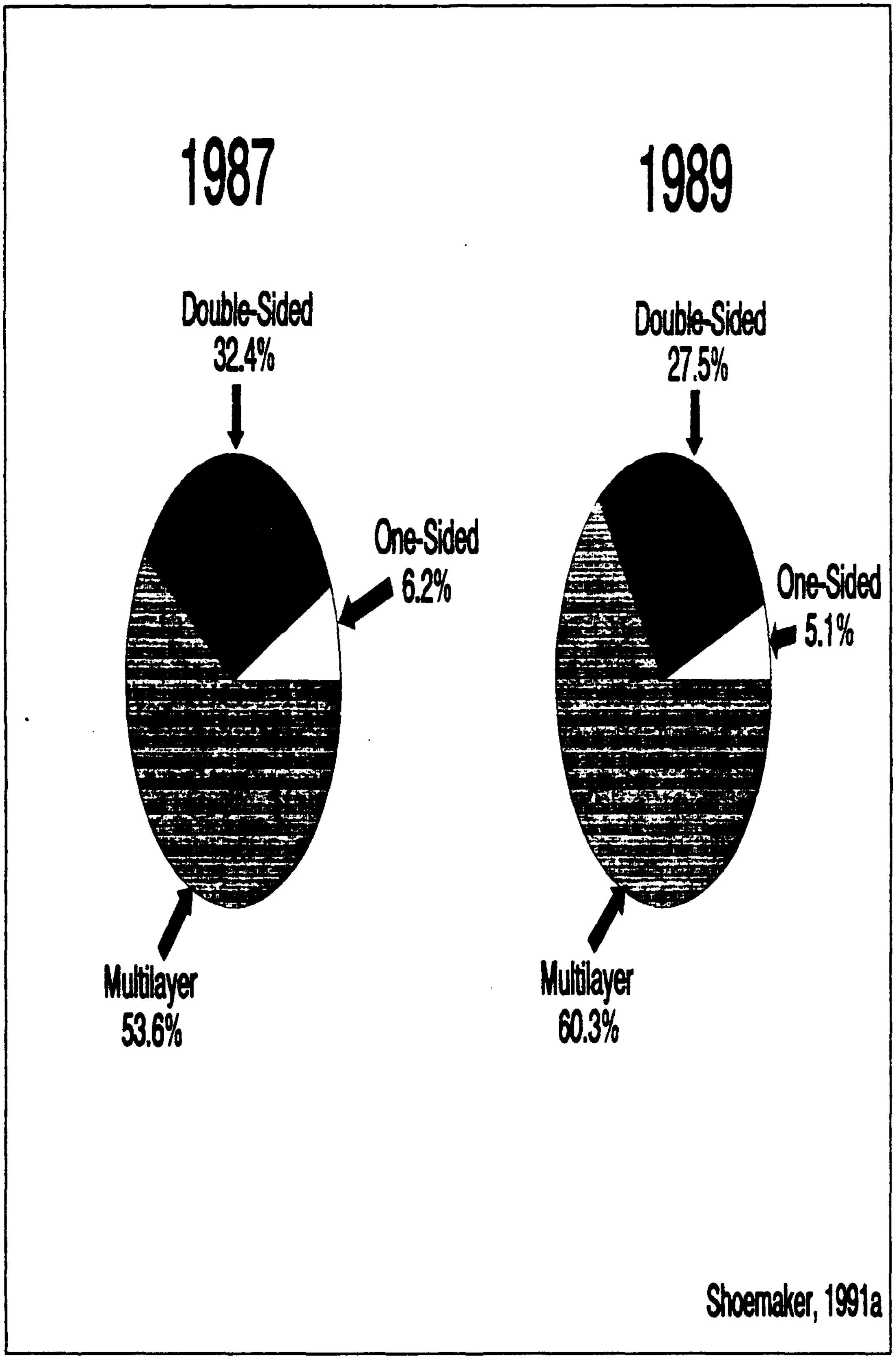

Figure 9. Growth in Domestic Multilayer Demand

Zycon has consistently outperformed the domestic U.S. board industry in sales growth for the last seven years (Shoemaker, 1991a). As shown in Figure 10, sales for the corporation have steadily increased since 1977. Like the rest of the industry, Zycon experienced a downturn in 1986 and 1987, but its losses were much less than the industry average. Beginning in 1988, sales markedly increased (Shoemaker, 1990). Despite rather flat sales growth for the industry as a whole, in 1991 Zycon's sales have increased by 29% (Shoemaker, 1991a). As shown in Figure 11, Zycon's percent of market share has also grown steadily (Shoemaker, 1990). In 1987 it was approximately 3.9%, and today it is about 6.7% of the industry Trade Association's database (which reflects roughly 40 percent of U.S. shipments (Shoemaker, 1990).

Marketing Strategy. Zycon has targeted their products almost exclusively toward the computer/peripheral market. As shown in Figure 12, 100% of their sales have been aimed at the top three market segments: computers/peripherals, communications, and industrial/instrumentation (Shoemaker, 1991a). Zycon manufactures boards for over 75 customers, and serves its customer base through direct sales (Shoemaker, 1991a). The company's major customers include Apple Computer, Compaq, Hewlett Packard, NCR, Silicon Graphics, Sun Microsystems, 3Com, and Unisys (Shoemaker, 1991a). In general, most of these customers desire long-term relationships with rigorous screening prior to qualification (Shoemaker, 1991a).

Most of Zycon's customers are considered high-technology producers, and as such they demand sophisticated boards that frequently require faster than average delivery times (Shoemaker, 1991a). Zycon does not try to low bid--their approach is quality (Gishi, 1991). It

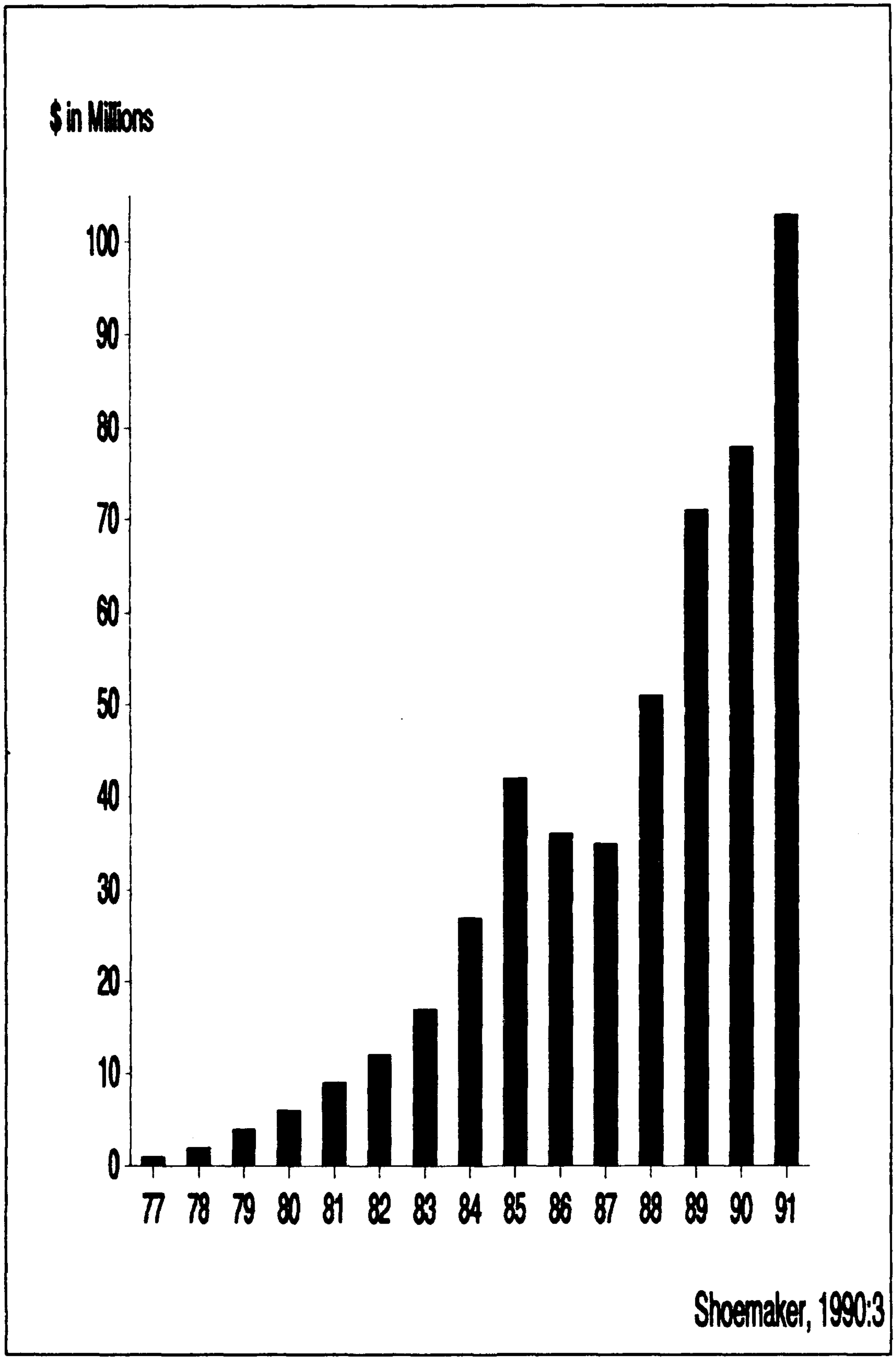

Figure 10. Zycon Corporation Sales

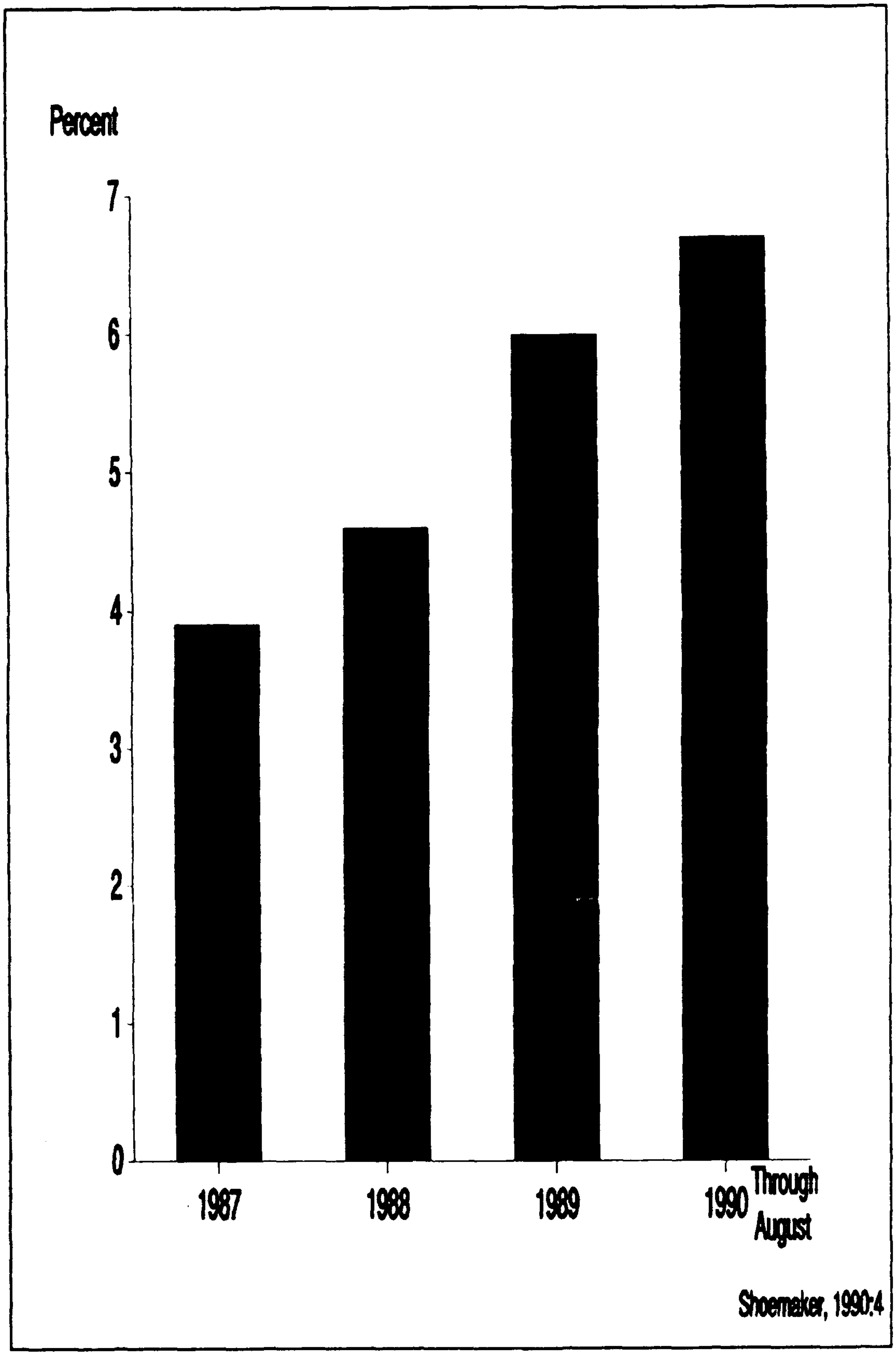

Figure 11. Zycon Market Share

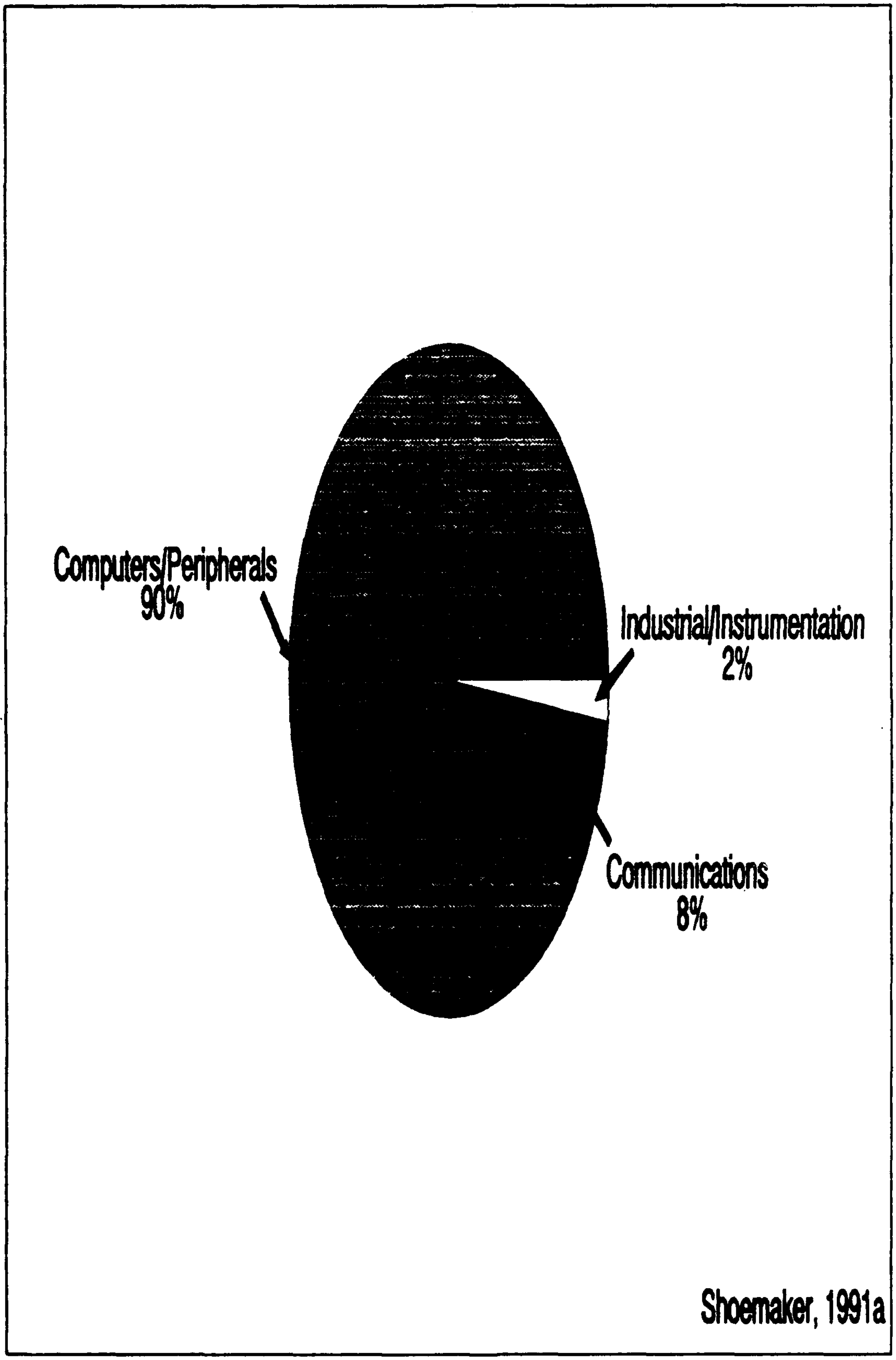

Figure 12. Zycon Target Market Segments

may take longer for their customers to get the product, but "they can count on it when it is received" (Gishi, 1991). The most important measure of performance for the company is due-date (full lots, on-time) (Shoemaker, 1991b). Reliability of delivering orders is the key to their business; however to be competitive, they still need to provide the product within a reasonable window of time (Shoemaker, 1991b).

Production Process. The Zycon production process can be classified somewhere between an intermittent process (a job shop) and a repetitive process. Although every job is a customer order, a number of products are repeat orders and many of the different products require very similar processing steps. To produce a multilayer circuit board, from incoming materials to shipping the final product, requires over 50 process steps. These steps include electrical, chemical, mechanical, and optical operations, plus testing at key points to ensure quality is built into the product (Zycon, 1991:1). By convention, the processes before drilling are collectively referred to as innerlayer, while operations subsequent to drilling are called outerlayer. Figure 13 is a flow chart of the major processes involved in Zycon's production operation, and each step is described below (Zycon, 1991:1).

1. Photo and Tooling. When an order is received, the engineering department prepares the photo tools necessary for manufacturing. At this stage numerical control procedures to control the drilling, routing, and testing equipment are defined.

2. Multilayer. When the work is for multiple layers, then this stage manufactures innerlayer cores with ground planes and circuit designs, then presses the cores together to form panels from which individual circuit boards will later be cut.

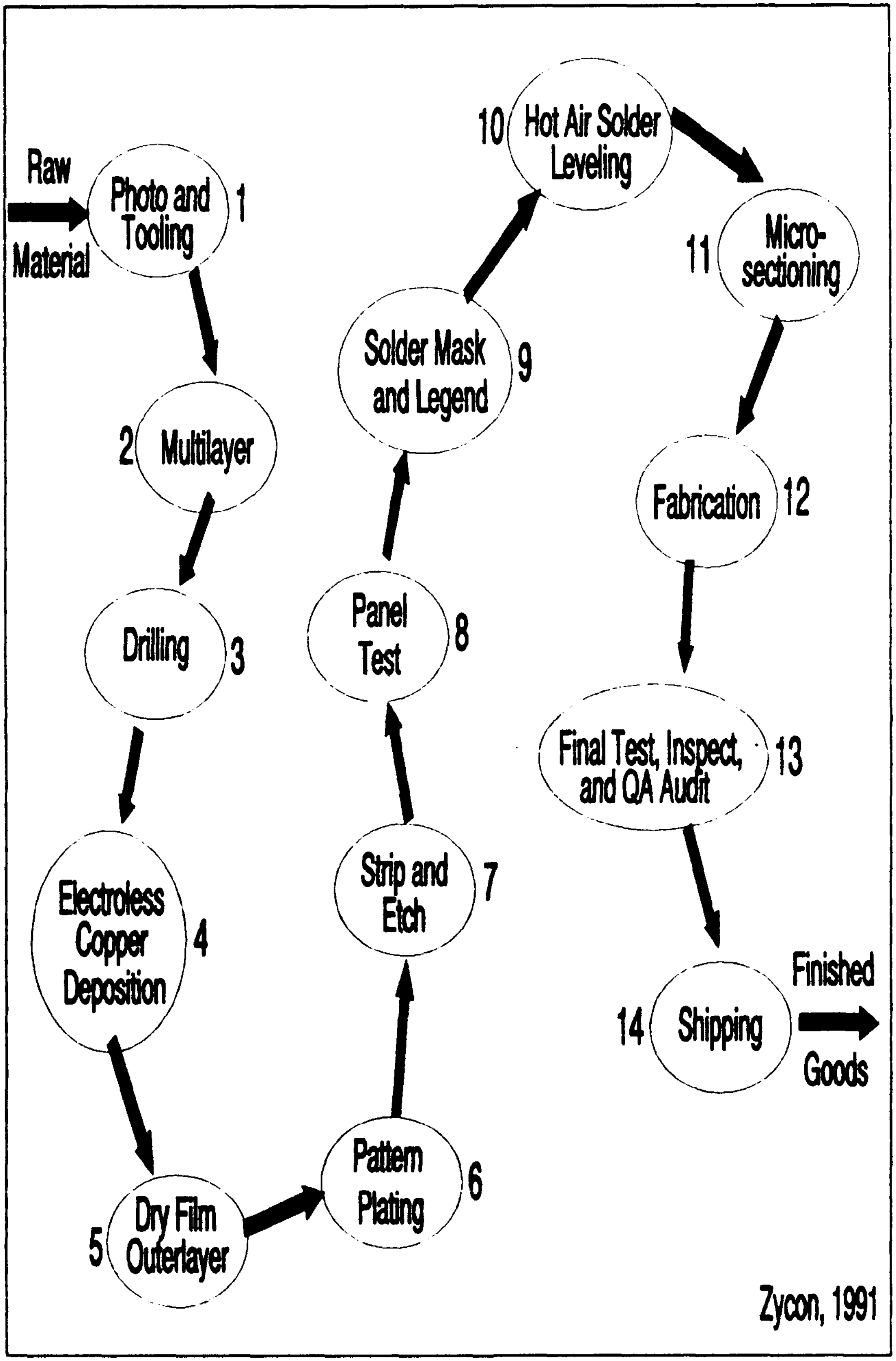

Figure 13. Major Production Processes and Flow

3. Drilling. Holes are then drilled through the board. These holes facilitate a continuous electrical path between the layers and provide holes for customers to mount components.

4. Electroless Copper Deposition. During this operation, the panels are dipped in a copper solution to deposit copper in the barrel of the drilled holes.

5. Dry Film Outerlayer. This operation coats the panel with a photoresist material, transfers the circuit board image to the panel using the photo tool artwork that was previously developed by engineering, then develops the photo image.

6. Pattern Plating. The panels are then electroplated in a copper solution to form electrical conductors on circuits and in holes.

7. Strip and Etch. This process removes the photoresist to expose copper, then removes unwanted copper, leaving the isolated conductors in the circuit pattern.

8. Panel Test. At this point, the panels are electrically tested for opens and shorts using a bed of pins custom designed to test conductivity.

9. Solder Mask and Legend. This stage coats the circuit pattern with a solder resist ink and prints a legend onto the panel to identify various components.

10. Hot Air Solder Leveling. The panel is dipped into molten solder, then blasted with hot air to even out solder coating on which customers will solder-mount components.

11. Microsectioning. This procedure cuts a cross-section from a board sampled from each lot, and examines its plated holes with a photomicrograph.

12. Fabrication. This operation cuts the panels into individual circuit boards.

13. Final Test and Inspect. This process consists of a final electrical test, a dimensional and visual inspection, and a quality audit to ensure compliance with customer requirements.

14. Shipping. At this point, Zycon packages, labels, and ships the product according to the customer's specifications.

TOC Implementation at Zycon

History. Zycon has historically been able to rely on the superior quality of their products: however, as competition increased, management began to realize they could no longer rely on quality alone--they needed an additional competitive edge (Gishi, 1991). Zycon believes the focused decision-making process offered by TOC has provided this competitive edge, and that TOC is a major reason for the company's consistent growth in market share (Shoemaker, 1991a).

Zycon was first introduced to TOC when a customer gave the marketing senior vice president (VP) (one of Zycon's co-founders) a copy of *The Goal* (Dunning, 1991). This individual was immediately impressed: however, (initially) few people shared his enthusiasm (Dunning, 1991). In early 1987, the President challenged top management to cut the cycle time in half within 18 months (Shoemaker, 1990). Faced with this demanding task, the executive managers sought new ways of doing business. A statistical process control team was formed in early 1987, consisting of the company VP's and several other key managers. In March and April of 1987, this team read *The Goal* and agreed to attempt a TOC pilot project (Shoemaker, 1990).

From April to August of 1987, Zycon conducted a test of TOC in the innerlayer department, and the results were dramatic (Shoemaker, 1990). Management chose this department because it included many of the processes (i.e., stripping/etching, testing, inspection, etc.) possessed by the production operation as a whole--it resembled a miniature production line (Gishi, 1991). In addition, this operation is located at the front of the production flow. At the start of the pilot, lead time for innerlayer ranged from 9 to 14 days (Gishi, 1991). After implementation, WIP levels decreased significantly, enabling managers to focus on the constraint (which for this department turned out to be an inspection process), and get this operation to faithfully follow a production schedule (Gishi, 1991). In line with drum-buffer-rope (DBR) theory, management instructed non-constraint resources to work only if material was available (Gishi, 1991). By monitoring the release of material and carefully scheduling the constraint, Zycon was able to markedly improve the performance of this department. Lead time for innerlayer was reduced to just three days (Gishi, 1991).

Excited by this success, the management team next concentrated its efforts on extending TOC to the outerlayer processes. Their efforts resulted in a 25 percent reduction in cycle time with a simultaneous increase in throughput; however, in January/February of 1988, the TOC implementation suffered a set-back. Feeling uncomfortable with the small amount of work in the shop, top management decided to increase the WIP inventory. The result was decreased throughput and quality, and increased cycle time. (Shoemaker, 1990d:8). Realizing the impact of this excess inventory, top management reversed its position; however, it

took four months for the company to return to the inventory levels of late 1987 (Shoemaker, 1990d:9).

In July 1988, the President and the marketing senior VP attended a 2-day introductory TOC class in New Haven (Shoemaker, 1990d:10). While in New Haven, the President asked the Goldratt Institute to train his top managers. Mr. Bob Fox came to Zycon and trained all first/second/third shift manufacturing managers, all department managers, several engineers (including the engineering management team), and the facility controller. This broad cross-section of employees comprised approximately one third of Zycon's managers (Shoemaker, 1991b). After this training, everyone was very excited about the application of TOC and committed to its implementation (Shoemaker, 1990d:11).

From August to December of 1988, the company closely followed TOC, especially with respect to the five focusing steps (Shoemaker, 1991b). During this period, the constraint wandered (Shoemaker, 1991b). For example, during November 1988, Zycon successfully booked more complex products (i.e., more multilayer work) into the plant. This change in product mix moved the constraint from the outerlayer shop to the innerlayer develop etch and strip operations. When management became aware of the new bottleneck, the engineering manager took measures to get a second line operating in the innerlayer department (it was originally designed for two lines, but only one was currently operating). Within 5-6 weeks, this second line was operational: however, until then, innerlayer was the constraint (Shoemaker, 1991b). The marketing VP then decided to bypass this constraint by accepting an order for 10,000 double-sided panels. If evaluated from a cost-

accounting viewpoint, this low-technology order would have been rejected; however, from a throughput perspective, since the double sided panels placed no extra load on the innerlayer operation, they resulted in "pure profit," above raw material costs (Shoemaker, 1991b). In TOC terminology, this order is called "free product." From a Zycon point of view, the theory was correct, and from the customer's perspective, Zycon was a hero (Shoemaker, 1991b). This example was one of the few opportunities that Zycon has had to do free product, but at points in time they do have free-product opportunities (Shoemaker, 1991b).

August through December of 1988 became the most profitable period in Zycon's history; however, the company experienced another major setback beginning in January of 1988, when a Director for Manufacturing was recruited from outside the company. Unfortunately, this individual refused to adopt the TOC philosophy (Shoemaker, 1991b). He had a very solid background in a related industry and had been very "successful" using cost-world manufacturing logic (Shoemaker, 1991b). His refusal to accept and use TOC concepts on the manufacturing floor, especially with respect to WIP levels, significantly set back Zycon's TOC implementation (Shoemaker, 1991b). Eventually, the individual sought employment elsewhere (Shoemaker, 1991b).

Zycon's failure to get the President involved at the outset was also a tremendous hinderance (Shoemaker, 1991b). Furthermore, the hiring of an individual who refused to adopt TOC significantly impacted the company's implementation efforts (Shoemaker, 1991b). From January of 1989 to October of 1989 the company's progress stagnated and actually reversed (Shoemaker, 1991b). For example, manufacturing undid much of the progress it had made with respect to WIP reduction, believing higher

inventory levels were necessary to protect against disruptions that might occur as a result of the relocation of the plant's outerlayer operations (Shoemaker, 1991b). Once the move was complete, the company then had difficulty getting their new automatic plating lines working properly. These new automated lines created more rework and scrap than the former facility's old manual lines. For a period, Zycon decided to operate the new lines in addition to the old ones, hoping to work the "bugs" out of the new line. Unfortunately, this policy also created additional scrap and rework problems (Shoemaker, 1991b). After two to three weeks, Zycon finally realized that running the new line was doing more harm than good, so it was shut down (Shoemaker, 1991b).

In August of 1989, realizing the importance of having all the key players involved in TOC, Zycon's President requested that Bob Fox return to Zycon and train all of the key managers as Jonahs (Shoemaker, 1991b). The President recognized that without the involvement of *all* key managers, TOC would not be successful (Shoemaker, 1991b). If only the mid-level managers are Jonahs, they will become frustrated if they identify potential improvements but are unable to make needed policy changes. Likewise, if only the top person is a Jonah, then the lower-level managers will not recognize significant opportunities (i.e., market outlets) and exploit them.

From October, 1989, to January, 1990, 12 of Zycon's top managers, including the President, completed the Jonah training course (Shoemaker, 1991b). This time, since all the key players were now Jonahs, the company finally began to focus on TOC, especially with respect to implementing buffer management (Shoemaker, 1990d:14). Since implementing TOC, the company's throughput is up 29 percent, inventory

is up only 1 percent, and operating expense is up 20 percent (due mainly to the fixed costs of opening a new 250,000 square foot facility). In addition, net profit is up 191 percent and inventory turns have improved by 30 percent.

The Manual Drum-Buffer-Rope. After the 1989/90 Jonah training session, the new Jonah group decided to try to force the constraint to be drilling and release jobs six calendar days before they were required for processing on the drill (the resource constraint buffer) (Shoemaker, 1991b). In addition, the drilling operation was planned for completion 18 days before the parts needed to be shipped (the shipping buffer) (Shoemaker, 1991b). Management's goal was to get total processing time down to 15 days when it was actually running closer to 29 days overall (Shoemaker, 1991b). They then tried to run the inventory out of the shop by carefully watching the content of the buffer-origins, marking the beginning of Zycon's drum-buffer-rope system (Shoemaker, 1991b). The Jonah group now began to recognize the difference between excess and protective capacity. To make their DBR system successful, the group realized they would have to identify what levels of protective capacity were required before and after the constraint, enabling them to control and maintain drilling as the strategic constraint (Shoemaker, 1991b).

Zycon is currently using this manual drum-buffer-rope scheduling system, and they have implemented many of the concepts of DBR into their present management information system (Gishi, 1991). Since the production process consists of only a straight line with one resource constraint, the company feels that they are able to see the impact of their implementation quicker than company's with more complicated process structures (Gishi, 1991).

Identifying the Constraints. Zycon's market desires full lots, on time. If the company can deliver accordingly, they believe they can sell anything, so the market will not be a constraint (Jo, 1991). By properly implementing and managing drum-buffer-rope, Zycon hopes to force the constraint to be internal, enabling them to control it (Jo, 1991). Even though the drilling operation is not commonly regarded as the system constraint, Zycon has artificially declared this operation as the strategic constraint (Gishi, 1991). Most of Zycon's managers regard the multilayer operation as the constraint; however, the drilling operation was declared the constraint because it is close to the front of the process, it has the most capital expenditure required to increase capacity, and the drill time itself can be very long depending on product type (Gishi, 1991). To maintain drilling as the constraint, the company has to supplement multilayer with outside capacity to ensure that it can do at least the amount of capacity available on the drill (Gishi, 1991).

Establishing the Buffers. Zycon's manual DBR system excludes the photo-engineering processes. As shown in Figure 14, rather than permitting the market to be a constraint, Zycon is attempting to establish the drilling operation as the strategic constrair'. The Jonah group decided (based mainly on judgement) on a resource constraint buffer for drilling of 6 days and a shipping buffer of 18 days. Each of the buffer-origins corresponding to these two time buffers is split into three regions, or zones. Zycon has established the constraint buffer so that material will arrive at the constraint 2 days before required for processing 100 percent of the time, and it will arrive an additional day earlier (3 days) 50 percent of the time (based on an agreed cycle time

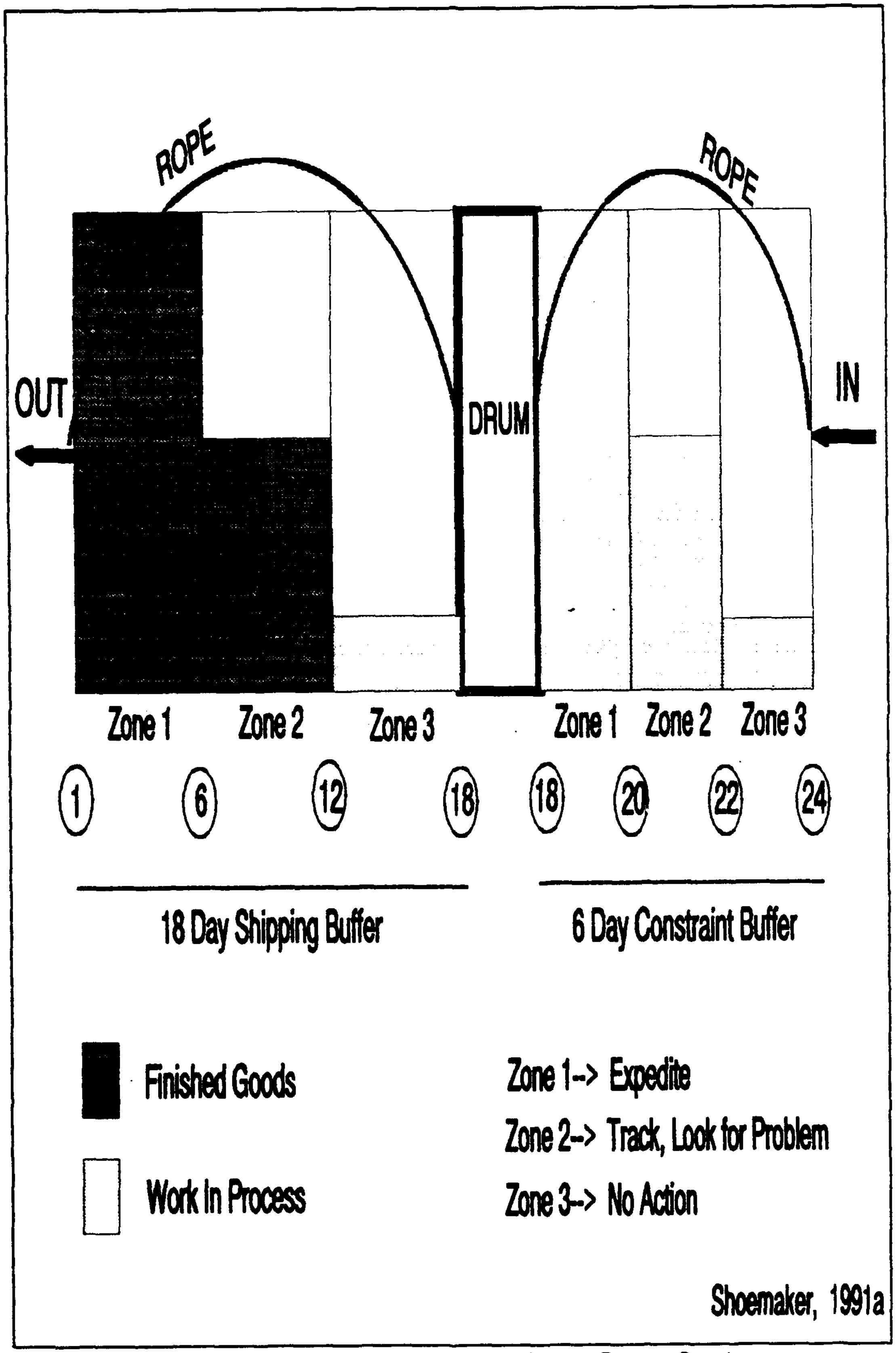

Figure 14. Zycon Drum-Buffer-Rope System

of 3 days for innerlayer) (Shoemaker, 1991b). Likewise, Zycon's shipping buffer is set up so that finished goods arrive at shipping 6 days early 100 percent of the time, and they arrive an additional 3 days earlier (9 days earlier than required) 50 percent of the time (Shoemaker, 1991b). In other words, Zycon has established the DBR system such that all of zone one and half of zone two will be full of finished goods (whether awaiting shipment or already shipped) (Gishi, 1991). Total *lead* time for Zycon's product is estimated to be 24 days (Gishi, 1991). The actual *cycle* time depends upon the WIP inventory level, and this time has ranged from 18 to 21 days over the last six months (Gishi, 1991). Management hopes to get the cycle time down to about 12 days or lower (Gishi, 1991).

Maintaining Control. Currently the company uses buffer location reports to schedule and track jobs through the system. Zycon uses several versions of the buffer report. The basic report identifies the sales order by Zycon ID, customer part number, revision, due date, quantity, balance due, and the source (Wright, 1991b). This report is sorted by department, so each manager gets a listing of jobs in his or her area, updated at the beginning of each shift (Madarieta, 1991). These reports show what jobs are located in each buffer zone, sequenced by due date (Gishi, 1991).

Production control maintains two dates: the "dock" date, the date the order is actually due to the customer, is not revealed to the shop-floor: and the "ship" date, is the date given to the shop-floor for completion of the order (Gishi, 1991). Rather than providing separate schedules for "hot," "normal," and "cold" items, production control produces only one schedule and manipulates the ship date to ensure that

the most urgent items are on top (Gishi, 1991). The schedule is sorted by ship date, and *production control* expedites jobs by artificially changing this date to ensure immediate items are listed first (Gishi, 1991). Rearranging the ship dates may result in a gap between the dock and ship dates, but this practice results in a correctly sequenced job "dispatch" list that provides consistent direction to the shop floor (Gishi, 1991). Each department simply processes according to the schedule, from top down (Madarieta, 1991). Shops are then judged strictly according to due-date performance (Gishi, 1991).

Implementation Problems. To date, Zycon has not been completely successful in its attempt to implement DBR. The company has certainly made tremendous progress; however, they have not yet been successful in establishing the required DBR buffers. The percentage of holes plugged in Zone 1 of the constraint buffer is currently only about 39 percent. For the shipping buffer, it is only about 55 percent (Shoemaker, 1991c:7;12). Zycon's management recognizes the need to fill the buffer-origins; however, for a variety of reasons, they have yet do so. They have successfully used an inordinate amount of expediting and overtime to remain on time to customers despite less than 100 percent plugging of zone 1 of the shipping buffer.

One reason for their failure to fill buffer-origins is the company's tendency to commit to ship all the available production capacity. When production is pushed to get everything they can out the door, the company has no extra time to start jobs early (and build the buffer). Despite one and one half years of trying and the fact that the plugging of the shipping buffer is the safest way to ensure on-time

deliveries, Zycon management has been unable to fill the shipping buffer.

The overcommitment problem is an example of numerous, unwritten policy constraints that continue to affect the implementation of TOC at Zycon. The unwritten, informal nature of these rules makes them very difficult to identify and even more difficult to overcome. An example of one such policy constraint was the company's long-standing policy to push production by directing the production of a certain number of panels per day. After the in-house Jonah training, most of the top managers, including the President, agreed to change this policy; however, there is still a tendency for managers to be evaluated according to panel count. Use of this measure assumes that the production rate at all departments must be equal, in effect treating all resources like the constraint.

Another problem relates to the fact the company has yet to completely undergo the requisite cultural change necessary for TOC. For example, shop-floor personnel tend to "panic" when they do not have enough material on hand. Management knows that reducing WIP is the key to lead time reduction, but they are still having problems communicating this idea. In fact, during the innerlayer pilot, as the level of work in process (WIP) was reduced, the managers found it necessary to put extra jobs in the system just to keep the people comfortable--to show them there was still enough work (Gishi, 1991).

Some managers still feel that there is a problem with respect to marketing. While they feel that marketing has come a long way, the perception exists that salesmen still fail to properly recognize the importance of seeking opportunities based on amount of dollars generated

per constraint processing hour instead of average panel price. On the other hand, marketing is having a difficult time managing bookings: as WIP levels come down, lead time decreases. As a result, the company receives more orders, which tends to increase the quoted lead time. Zycon's customers have a hard time understanding the difference between lead time and cycle time, so they expect the lead time to be very short regardless of the number of orders in the plant (Dunning, 1991). This problem highlights the fact that Zycon does not possess enough protective capacity for many resources.

Benefits of TOC. The improvements resulting from Zycon's DBR implementation clearly overshadow these problems. Full-lot, due-date performance before TOC was only about 30-35 percent; however, now it is about 80% (Gishi, 1991). Before TOC, Zycon's production system contained jobs as much as several weeks overdue--now they are at most a few days late (Gishi, 1991). Before DBR implementation began, it was the responsibility of individual scheduler's to get "their" jobs through the system (Gishi, 1991). Management hoped that conflicts between jobs for a particular resource would "work themselves out;" however, normally the scheduler with the most clout got his or her job through (Gishi, 1991). Before TOC, Zycon also had an "army of expediters" running around the clock; now they only have two (Gishi, 1991). Now the manager of each department gets a list at the beginning of each shift that identifies exactly what jobs to run in what sequence, and they just follow it (Gishi, 1991). According to one manager at Zycon with significant experience producing PCBs in MRP environments, MRP seemed to work fairly well, but the TOC control system is clearly superior (Franzino, 1991). Probably the most important benefit is the highly-

focused nature of the complete management team, resulting in the ability to execute a straight-forward game plan (Shoemaker, 1991b).

DISASTER™ Implementation

Need for DISASTER™. Zycon now recognizes that they have three potential constraints: the outerlayer department (drilling), the innerlayer department (multilayer), and the front-end photo-engineering process (Shoemaker, 1991b). It is very difficult to keep the panel production matched between innerlayer and outerlayer processes (Shoemaker, 1991b). Some panels coming to outerlayer are very simple and may require only one hour for a drill "run:" however, other panels may require almost five hours for a drill run, depending upon the sophistication required (thickness, layers, hole sizes, density, etc.) (Shoemaker, 1991b). Likewise, some orders require much more work in the innerlayer department, but the amount of work depends upon different factors (i.e., number of layers). Zycon has several alternatives for balancing workload in these areas. For example, they can supplement the innerlayer department with outside subcontracting, or they can pursue more low technology business (such as double-sided boards that require no innerlayers--they have circuits only on the top and bottom of the panel and thus require no work in innerlayer) (Shoemaker, 1991b). The point is that Zycon can supplement those points in time when they are operating close to maximum available capacity in innerlayer with more double-sided work or they can subcontract some of the innerlayer workload (Shoemaker, 1991b).

The third interactive constraint is the front-end engineering of the board, development of the engineering tools necessary for the lay-up and process capability during production (Shoemaker, 1991b). Regardless

of whether an order requires 5 or 500 boards, if it is a new order it must go through this process. The company has situations where innerlayer and outerlayer have the capacity to begin work, but they cannot do so because engineering was not able to complete the necessary tooling in time (Shoemaker, 1991b).

Due to the types of workers (skills) and machines required in the photo-engineering area, this constraint is very difficult to break (Shoemaker, 1991b). This department is doing a lot of work to automate their process and thus reduce the cycle time and defect rate; however, marketing still believes that the company is not competitive with respect to the speed of the front-end operations (Shoemaker, 1991b). Despite these facts, management is still reluctant to invest in the amount of protective capacity necessary to respond to fluctuations in mix of new versus repeat orders--it is extremely expensive (Shoemaker, 1991b).

Instead of investing in more protective capacity in these areas, Zycon is trying to manage the interactivity between these constraints (Shoemaker, 1991b). Regulating the mix of repeat orders (which do not require engineering) versus new part numbers is one way. Another way to do this is to market intermediate products: products that move through only one of the interactive constraints, allowing the company to balance the load on each of the constraints while limiting interactivity between them (Shoemaker, 1991b). As shown in Figure 15, Zycon has several possibilities. The company could offer their laser plotters and engineers to do only a portion of the photo-engineering operation, such as artwork and digitizing, for other companies. Another possibility is to offer the engineering department as a service bureau to smaller

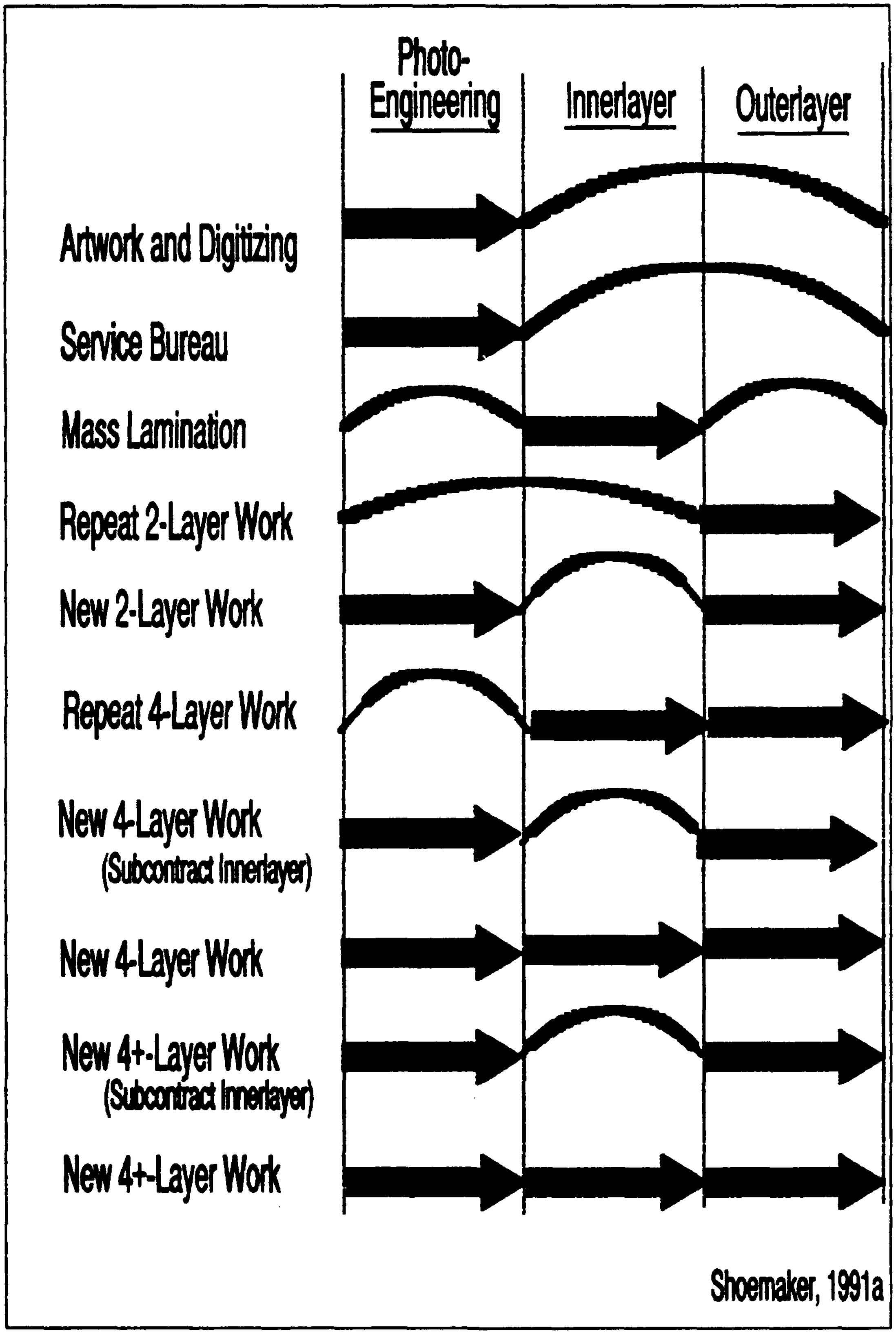

Figure 15. Managing the Constraint with Intermediate Products and Product Mix

companies. In both cases, orders would affect only the photo-engineering department. With respect to innerlayer, the company could market just their mass lamination operation. Finally, to control the capacity of the outerlayer department, Zycon could control the mix of repeat versus new work and double-sided versus multiple layer boards to balance the load on the interactive constraints (Shoemaker, 1991b). Use of intermediate products and careful management of product mix would justify keeping (or adding) additional protective capacity necessary in these departments (Shoemaker, 1991b). If the market peaks, Zycon could simply refuse orders for these intermediate products, thus freeing more capacity to meet regular demand (Shoemaker, 1991b).

The primary reason Zycon's Jonah group agreed to implement *DISASTER*™ is because of the interactivity between these constraints and the difficulty they experienced in matching capacity (Shoemaker, 1991b). The theory of constraints advocates breaking interactive constraints; however, in Zycon's case, breaking the constraint would be enormously expensive, so the company instead intends to use *DISASTER*™ to manage the interactivity (Shoemaker, 1991b). Due to its ability to authorize multiple drums, Zycon expects *DISASTER*™ to be able handle their interactive constraints--something that is very difficult or impossible to do with a manual DBR system (Shoemaker, 1991b).

One of the initial implementation obstacles was the development of a tracking system that included photo-engineering. Under the company's manual DBR system, engineering was not included as part of the production system (Shoemaker, 1991b). During the early stages of implementation, the management information system (MIS) department worked to establish a network for the tooling loop that could be

downloaded into *DISASTER™* (Shoemaker, 1991b). In addition, for the last several months the MIS department has been working to get the company's data into a form suitable for *DISASTER™* (Shoemaker, 1991b). The company has now overcome most of the data and job tracking problems, and they are now very close to placing *DISASTER™* on the shop floor.

Zycon Databases and Job Tracking. Zycon's database is segmented into two systems that evolved as the company evolved: one in manufacturing, and one in engineering (Wright, 1991b). The engineering side is a UNIX-based system, and the manufacturing system is on a PRIME minicomputer (Wright, 1991b). The PRIME side has traditionally handled the business applications: accounts payable and receivable, orders, and most recently job tracking through the facility, most recently with bar coded data entry from the shop floor (Wright, 1991b).

The company has had a batch-update work order tracking system since 1982, and for the past year and a half they have been tracking movement of the jobs through the shop on-line. Zycon assigns a work order to each job that moves through the production plant (Wright, 1991b). Various terminals are located throughout the plant and used to update the status of each job as it moves through the plant (Wright, 1991b). Each of the terminals is located at key locations in the process and operators input status as the jobs flow through their area (Wright, 1991b).

The capability to track all information is a key to the success of *DISASTER™*. *DISASTER™* is dependent upon the job order transfer system that monitors jobs as they progress through the plant. Without a reliable tracking system, *DISASTER™* will not be effective (Jo, 1991). The determination of yield (how much of the product is lost during the

production process) is very important in Zycon's business (Jo, 1991). Typically, products experience an average yield of about 88 percent, but it varies significantly according to product type (Jo, 1991). In the past, Zycon has had difficulty in accurately predicting yield (Jo, 1991). To realize the full potential of *DISASTER*™, Zycon must be able to determine yield fairly accurately, enabling them to know what is out on the shop floor (Jo, 1991).

Originally, Zycon's on-line job tracking was only designed to track "good" product, with scrap recorded in a different system using a different, manual procedure (a paper log was maintained and an individual performed batch data entry into a separate database maintained on the PRIME system) (Wright, 1991b). Within the past six months, the company has integrated the scrap tracking system with the rest of the system. Whenever material is moved, the operator is now queried for the number of pieces (panels) moved, and the system compares this entry to what it believes should have been moved (Wright, 1991b). If there is a difference, then the system requires the operator to enter the amount of scrap and a code to identify the reason for the scrap. The operator is now forced to account for any disparity between what the system thinks should be moved and what is actually input, enabling the system to now track movement of *all* material (Wright, 1991b).

About two and one half years ago, Zycon began producing a traveler to accompany each order through the plant. These travelers contain the routing for the job and identify what specific processes are required for a particular lot (Wright, 1991b). Since the engineering side developed and maintained all of the required information on the UNIX system, this system was used to electronically produce the traveler

(Wright, 1991b). As Zycon began to develop the PRIME-side tracking system, they quickly realized that they needed much of the same information maintained on the UNIX system (Wright, 1991b).

This problem was the primary obstacle during the early stages of *DISASTER*™ implementation: the company used two different databases that needed the same information, but there was no easy way for these systems to communicate (Wright, 1991b). Zycon's intermediate solution was to use a communications program, such as *CROSSTALK*™ or *PROCOMM*™, to download the files from one machine and upload them into the other (Wright, 1991b). Unfortunately, the uploading and downloading time to get the files required for *DISASTER*™ was prohibitive (not only the time for flow between the two databases, but also the time to download to *DISASTER*™'s microcomputer) (Wright, 1991b). The size of the *DISASTER*™ project data set files is 600-1000 lines for the order file, 50-60 lines for the resource file, 300-400 lines for the raw material file, 18,000-20,000 lines for the station files, and 20-24,000 lines for the arrow file (Wright, 1991b). Zycon's station and arrow files are each well over 1 megabyte (Wright, 1991b). Due to the size of the files, downloading was (initially) an extremely slow process. A complete set of *DISASTER*™ files for one scheduling run often took two to two and one half hours to download (Wright, 1991b).

Zycon recently installed a new communications protocol that allows machines with dissimilar operating systems to talk to one another on a real-time basis (Wright, 1991b). This software significantly speeds up the process. For example, a file that once took one hour to download under Zycon's original procedure now takes only about one minute (Wright, 1991b). Use of this software has solved the file transfer

problem, reducing the data collection task to a quick and simple downloading procedure (Wright, 1991b).

Creating the Project Data Set. Since the bulk of the information required for the project data set was located in the PRIME computer, Zycon chose this system as the vehicle for collecting and converting the data (Wright, 1991b). The UNIX programmer uses a utility to create a bill of material file and a raw material file and then transfers these files to the PRIME system (Wright, 1991b). The raw material file is transferred in the format required by *DISASTER*™ (Wright, 1991b). A utility on the PRIME system then uses these two files and other available information to create the project data set, starting with the order file and progressing in the sequence suggested in the *DISASTER*™ software documentation (Wright, 1991b).

Zycon makes several manipulations to the data to account for unique characteristics of the Zycon environment (Wright, 1991b). For example, production uses transfer and process batch sizes of 48 panels (Wright, 1991b). When an order is received, it is broken into work orders (batches) consisting of 48 panels; however, for *DISASTER*™, Zycon did not want to model work orders (Wright, 1991b). The company does not get credit for shipping work orders, but rather for shipping full lots (Wright, 1991b). To get around this problem, Zycon chose to model the production process as a straight line until the products reach finished goods. At this point, the order is then split into separate work orders, and then merged back into the sales order as finished goods (Wright, 1991b). For example, consider Figure 16, describing two orders by the same customer for the same product, but with different delivery dates (Wright, 1991a). Zycon assigns the same sales order number to

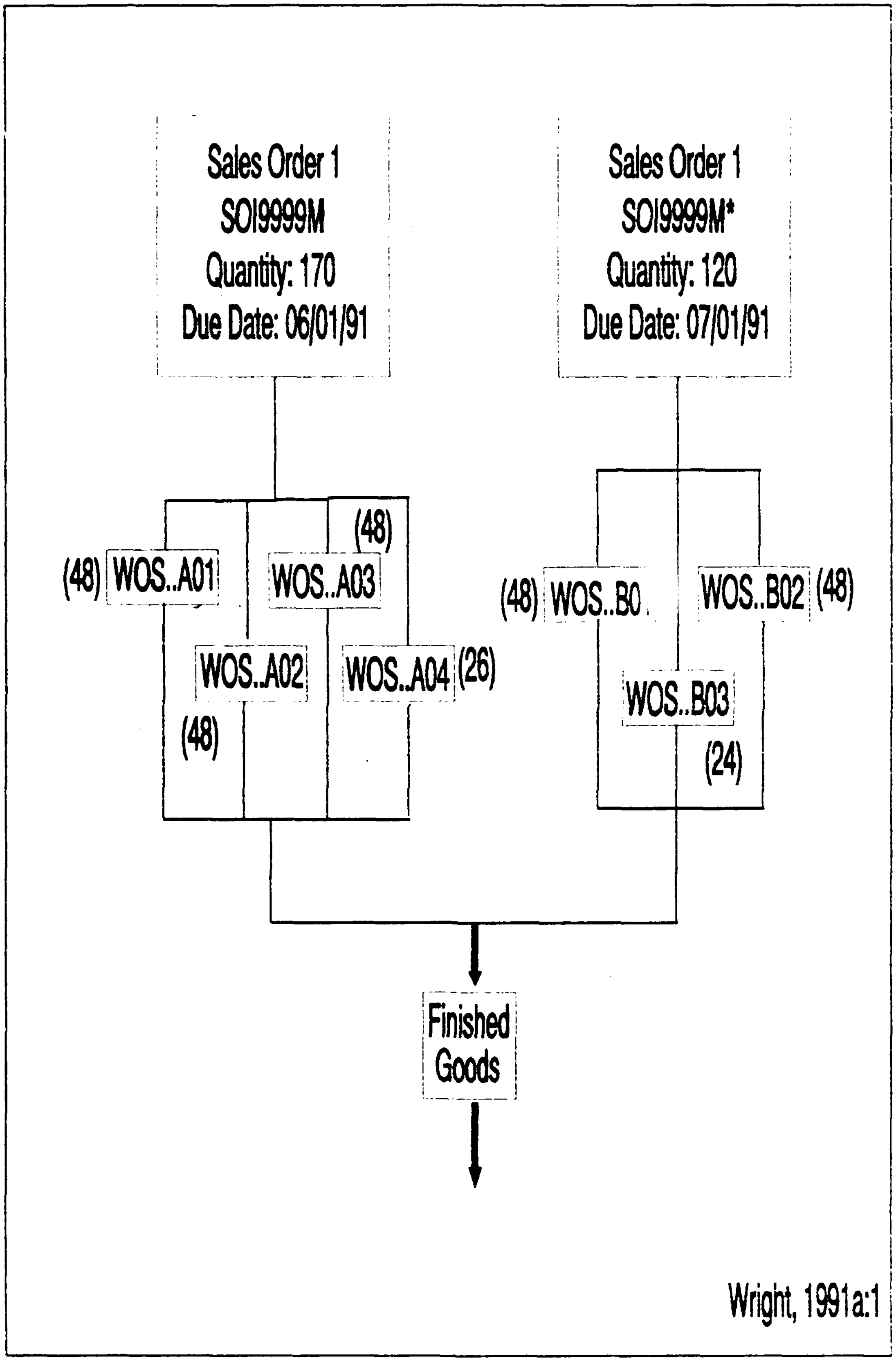

Figure 16. Modeling Logic for *DISASTER*

both orders, but splits each into separate work orders with unique identifiers (i.e., A01..A0*n* versus B01..B0*n*). Using this logical structure, *DISASTER™* is able to handle multiple orders for the same product. In addition, splitting the products into separate work orders permits *DISASTER™* to schedule backward in chunks of 48 (or fewer) panels (Wright, 1991b). Each customer order is not considered complete until the appropriate number of panels has been completed (Wright, 1991b).

Aside from this peculiarity with the order file, the other files are basically the same as with any *DISASTER™* application (Wright, 1991b). The resource file is strictly a listing of Zycon's resources. Zycon rarely uses the default calendar, instead specifying resource-specific calendars for almost every resource (Wright, 1991b). As noted above, the raw material file is produced on the UNIX system in the proper format (Wright, 1991b).

The bill of material (BOM) file from the UNIX system is provided in ASCII format, then is converted into a dynamic file (on the PRIME) that can be indexed very quickly (Wright, 1991b). Zycon has generated a utility that takes information from the BOM file and the previously-created order file and creates the station and arrow files (Wright, 1991b). This utility goes through and reads each sales order in sequence, then simultaneously creates the station and arrow files (Wright, 1991b).

Zycon has some resources that are used differently at different points in the process (Wright, 1991b). For example, every job goes through drilling; however, a limited number of boards also cycle back through drilling a second time (towards the end of the process) (Wright,

1991b). In the routings file, these two processes are given unique resource identifiers, even though they use the same resource (Wright, 1991b). *DISASTER*™ then regards both resources separately (Wright, 1991b). Since such examples are somewhat rare, they are taken care of with a mass edit to change the identifiers for the second process (Wright, 1991b). At this point, the files are then sorted and are ready for downloading (Wright, 1991b).

The Outputs. The preceding methodology is pretty well-established (Wright, 1991b). Zycon has only recently begun to experiment with the various outputs (Wright, 1991b). Most of the standard output files will not be used by Zycon in the format provided (Wright, 1991b). Instead, Zycon plans to take information from these files and incorporate it into their present buffer management reports (Wright, 1991b). Figure 17 displays the overall data flow for Zycon's *DISASTER*™ implementation.

Most of the information used by Zycon is contained in the SD1 output file, the schedule for the constraints (Wright, 1991b). This file gives a schedule, starting at time zero of the schedule horizon, for each resource constraint unit (Wright, 1991b). The standard SD1 report has much more information than strictly required by Zycon's shop-floor personnel, so information will be selectively extracted from this report and inserted into their standard buffer report (Wright, 1991b). This report contains the sequence of the work to be processed on the constraint, the sales order requirement, the Zycon ID, the quantity, the date and time that it ideally should have started, the date and time it is scheduled to start/finish, what particular constraint unit the work is to be put on, the run time for the particular batch (Wright, 1991b).

The only information necessary for the shop-floor personnel

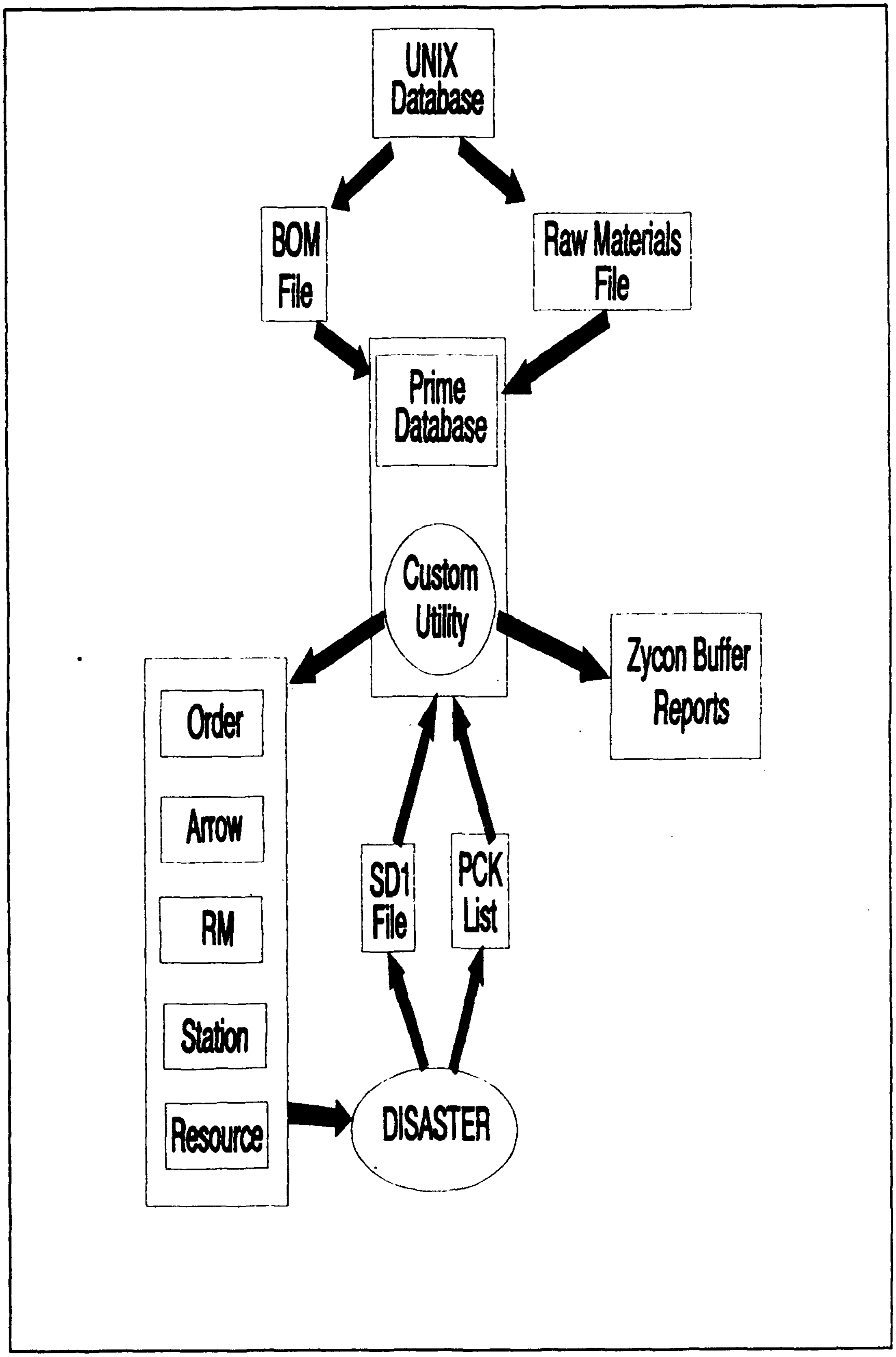

Figure 17. Zycon Data Flow

(operating the constraint), is the priority/sequence of the various orders, some identification of the part to be worked on, the work order, and the quantity (Wright, 1991b). Zycon has generated a utility to extract only this information and reformat it into reports for the shop floor (Wright, 1991b). They now have several sample reports that are being considered (Wright, 1991b). If desired, instead of scheduling the constraint by department, each individual constraint unit can be scheduled separately (Wright, 1991b).

The material release points at the front of the process are scheduled according to information contained in *DISASTER*™'s Pick list output file (Wright, 1991b). *DISASTER*™ also provides late order listing information. Using this option, managers can obtain a listing of all late orders, showing the scheduled due date and the date *DISASTER*™ predicts they will ship (Wright, 1991b). To date, Zycon is not attempting to utilize this information, but in the future Zycon will likely integrate it into their present buffer report (Wright, 1991b).

Problems/Potential Problems. Zycon has experienced several problems that have hindered *DISASTER*™ implementation efforts. Each of these problems was relatively minor, and each has been overcome. One limitation of *DISASTER*™ is its failure to recognize a shorter process time for less than full lots (Wright, 1991b). For example, the required number of panels often results in some work orders for the full 48 and a significant "left-over" number of panels (i.e., 23 or 24) to complete the customer order (Wright, 1991b). *DISASTER*™ will assign a process time on the constraint for this last work order equal to the process time for a full 48 panel lot, even though the actual time might be significantly shorter (Wright, 1991b). This problem could be solved by

providing an algorithm to define the batch process time (as opposed to the present method of allowing only a straight number to be input) (Wright, 1991b). Zycon has asked The Goldratt Institute for a revision to account for this problem (Wright, 1991b).

Another problem Zycon encountered is that when *DISASTER*™ goes through the subordination process, it automatically schedules any material that is not currently at the buffer-origin to arrive in a period equal to two thirds of the resource constraint buffer (4 days) (Wright, 1991b). Often, the material arrives much earlier than four days, but it must still wait for processing based on the schedule (Wright, 1991b). One solution is to schedule more frequently, but that would be a drain on resources (Wright, 1991b). A work-around the Goldratt Institute developed is to create a phony station for the constraint and to give this station infinite resources and a zero process time. As a result, all material is placed at this station rather than at the drill (Wright, 1991b). During subordination, *DISASTER*™ always displays a warning message stating that there is nothing in the buffer; however, when it schedules, it starts with an empty resource buffer, then places everything in proper sequence (Wright, 1991b).

A problem only recently encountered by Zycon relates to the effect the length of time selected for the scheduling horizon has on the process time and the output reports (Wright, 1991b). Marketing wants a report with a running tab of the layers of load on each department so that they can balance the load (Wright, 1991b). *DISASTER*™ has all this information and the MIS department can easily use a utility to format and extract this information; however, marketing wants a 3-month time

horizon (Wright, 1991b). Zycon's normal window is only about one week (Wright, 1991b). A normal scheduling run results in a stack of output about one quarter inch thick and takes only about two to three minutes (Wright, 1991b). For a 3-month horizon, it takes about 30 minutes and the output is about 2-3 inches thick (Wright, 1991b). Furthermore, it can take as much as 3 hours to convert the *DISASTER*™ output files into formatted reports (Wright, 1991b). To accommodate marketing, the MIS department plans to continue running the normal production schedule for a 1-week horizon, but also run a special marketing schedule (using a 3-month horizon) once each week (Wright, 1991b). This policy will improve marketing's ability to schedule and manage load without draining too much of Zycon's resources.

VI. Conclusions and Recommendations

Chapter Overview

This chapter provides a summary of conclusions and recommendations for this research. First, this chapter provides a brief summary of the research objectives by reviewing the investigative questions. Next, the major conclusions discovered during the research are identified. These conclusions relate to the problems and criticisms of MRP-based systems, the potential benefits of *DISASTER*™, and obstacles organizations implementing *DISASTER*™ (particularly AFLC) are likely to encounter. Finally, this chapter concludes with recommendations for future research.

Summary of Research

The primary objective of this research was to investigate the characteristics of the *DISASTER*™ scheduling system. Seven investigative questions were proposed in Chapter I, *Introduction*, to guide this research. Before analyzing *DISASTER*™, however, it was necessary to build a foundation of knowledge concerning current manufacturing planning and control systems. The first five questions established the basic foundation for the research:

1. What is the basic premise behind materials requirements planning (MRP)?

2. What is manufacturing resource planning (MRP II) and how does it differ from MRP?

3. What is the planned approach for DMMIS?

4. What are potential problems and limitations of MRP-based systems?

5. What TOC concepts form the basis of the *DISASTER*™ scheduling system?

Chapter II, *The Review of the Literature*, answered these questions by examining current manufacturing planning and control systems. This chapter began with an overview of materials requirements planning (MRP)-based planning and control systems. This section outlined the basic logic behind MRP systems, reviewed manufacturing resource planning (MRP II) and how it differs from standard MRP, described how MRP II will be applied to DMMIS operations, and identified potential problems with MRP-based systems. Next, due to the similarities between just-in-time (JIT) manufacturing and the theory of constraints (TOC), JIT was briefly introduced. Finally, the research provided a more in-depth review of key principles of TOC, in particular drum-buffer-rope and buffer management, upon which the *DISASTER*™ scheduling system is based.

Examination of these areas provided the background necessary to permit comparison of the operation and capabilities of *DISASTER*™ to traditional manufacturing planning and control approaches. Having established this foundation, the research next focused on answering the next investigative question: what are the specific characteristics of the *DISASTER*™ system and how does this system differ from conventional scheduling approaches? Chapter IV, *Analysis of DISASTER*™, focused on the operation and capabilities of the software package. This chapter was not intended to provide detailed instructions for its use. Instead, this chapter examined the conceptual basis for its operation, including the criteria for a good schedule, and *DISASTER*™'s decision process and scheduling logic/procedure. Next, the characteristics and operation of *DISASTER*™'s major software modules, CALENDAR, NETGEN, and SCHEDULE,

were examined. The required data flow was identified (development of the project data set by NETGEN and the calendars by CALENDAR) was outlined, the major steps used by the SCHEDULE module during scheduling were discussed, and the primary system outputs were identified. Finally, this chapter identified the potential benefits that might be realized from *DISASTER*™.

Once the characteristics of *DISASTER*™ were sufficiently examined, the research then focused on the final investigative question: what are the requirements for and the potential problems/obstacles inherent with implementation of the *DISASTER*™ system? To address this question, the researcher conducted a single, holistic case study of a commercial circuit board manufacturer, *The Zycon Corporation*, that is currently implementing *DISASTER*™. Unfortunately, the recent release date of the software (February 1991) limited the potential candidates for the analysis; therefore, the single case study design was necessary. Analysis of efforts by *The Zycon Corporation* represented a unique, first-time opportunity to investigate implementation of the software.

Conclusions

MRP Concerns. The literature presented in Chapter II clearly indicates the presence of growing concern among operations management experts regarding the use of MRP-based systems for scheduling and controlling manufacturing operations. While numerous concerns were identified, some of the major criticisms of MRP involve its failure to 1) properly address capacity limitations, 2) account for disturbances on the shop floor, and 3) identify and manage constraint resources.

MRP's assumption during schedule generation that resources possess "infinite capacity" is a major cause of unreliability of MRP-produced

schedules. MRP systems do not schedule consistently backward in time while considering and resolving capacity limitations. Instead, the MRP method assumes no limitations during schedule development, and then uses a capacity requirements planning module to try to correct for capacity problems *after* schedule development. This practice, coupled with the length of time it takes for generation, introduces some question as to the realism of any MRP-generated schedules. In addition to MRP's failure to consider capacity limitations during scheduling, it also fails to address the impact of disturbances that will inevitably disrupt shop-floor operations. This failure to account for the cumulative effect of statistical fluctuations occurring among dependent events during schedule development increases the likelihood that MRP-generated schedules will be unrealistic. Finally, a major disadvantage of MRP appears to be its failure to recognize and concentrate on bottleneck resources. MRP's attempt to schedule and control *all* resources results in problems such as excessive data and reporting requirements that greatly increase the management burden. In short, MRP systems attempt to manage so many resources and track so much data that they often become unmanageable, especially for very large systems.

These problems are certainly applicable to AFLC's use of MRP II for its Depot Maintenance Management Information System (DMMIS). There is little doubt that the switch to MRP II will be a definite improvement over the command's current system; however, given the advent of new manufacturing philosophies and the rising concern over MRP logic, MRP II may no longer be the *best* solution. The unique characteristics of AFLC's maintenance environment (versus a normal manufacturing operation), introduce additional concern regarding the likelihood of

DMMIS success. As noted, some reservation exists as to the ability of MRP II to provide realistic schedules. The increased uncertainty inherent in maintenance operations only increases these concerns. Furthermore, higher levels of uncertainty will likely result in material and capacity plans that are only estimates of the actual requirements: therefore, generation of schedules will need to be performed more often. Even with standard MRP II systems, run time is often significant, and with DMMIS, schedule generation will likely require even more time. If DMMIS is unable to generate reliable schedules in a timely manner, the ability of managers to perform what-if analyses will be significantly reduced.

Potential Benefits of DISASTER™. Although the application of *DISASTER™* has, to date, been very limited, the TOC concepts upon which the software is based (drum-buffer-rope scheduling) have been successfully applied in many commercial manufacturing companies. The review of the literature, the analysis of *DISASTER™*, and the case study of the application of the system at Zycon highlighted several key benefits that may result from its use.

The fact that *DISASTER™* focuses on the constraints not only permits the system to maximize throughput, but also it produces other key advantages. One of the major obstacles with MRP II systems is the need to obtain and maintain enormous amounts of data. Instead of attempting to obtain information and track the activities of all resources, *DISASTER™* focuses primarily on constraint resources. This fact enables managers to concentrate only on resources that significantly affect the operation, and the likely result will be improved management efficiency. The use of *DISASTER™* will also limit

the amount of data required, and then only require that this small subset of data (associated with the constraint) be accurate. In addition, unlike MRP systems that simply require that all data be "95 percent accurate," whenever a decision is required that involves information about the constraint, *DISASTER*™ presents this information for verification.

Another major advantage of *DISASTER*™ is that it recognizes and accounts for capacity limitations and disturbances during schedule development. *DISASTER*™ Schedules strictly backward in time, and the program addresses capacity problems for each resource before moving to an earlier date. In addition, the system uses an estimate of the level of disturbances on the shop floor to establish appropriate levels of protective inventory and protective capacity needed to shield against disruptions. These characteristics will likely result in schedules that are much more realistic than those produced using traditional scheduling methods.

The fact that *DISASTER*™ structures the data in a concise format and then maintains everything in memory permits the program to produce schedules much faster than traditional methods. While this fact may not seem particularly significant, when one considers that schedules are usually the basis for answering managerial what-if questions, then it becomes apparent that generation speed can be very important. It appears that *DISASTER*™ will enable managers to produce schedules very quickly, thus enabling the system to answer multiple what-if queries within a short time frame. Furthermore, given the high level of uncertainty present in depot manufacturing and the probable need to

rerun schedules more often, the fast processing speed of *DISASTER™* becomes even more significant.

In addition to these advantages, *DISASTER™* is simple and flexible. Once the user understands the basic TOC concepts, the program is very easy to use. It is entirely menu-driven, and it allows the user to mover through the menus to obtain as much or as little data as desired. Furthermore, the system appears to be generally applicable to many organizations that are currently using other scheduling systems. The program requires its inputs to be supplied in five generic ASCII files that can easily be generated by most conventional computer systems. For MRP systems in particular, all of the required data (plus more) has likely already been collected, so most organizations need only be concerned with developing procedures to locate and format it.

In summary, the fundamental new management philosophy (TOC) upon which *DISASTER™* is based appears to provide significant opportunities for improvement over conventional scheduling approaches. Unlike MRP systems that concentrate on optimizing local operations, *DISASTER™* focuses on maximizing the throughput of the system as a whole. Furthermore, by recognizing and removing conflicts between constraints within the system, accounting for capacity limitations, and considering the impact of statistical fluctuations and disturbances, it appears that *DISASTER™* can produce realistic, highly reliable schedules.

Implementation Obstacles. Implementation of *DISASTER™* in any organization is certainly not be an easy undertaking. The use of *DISASTER™* requires that the implementing organization undergo a drastic cultural change: from the cost world to the throughput world. Few companies have undergone such change--it is difficult for any company to

make a significant change in its fundamental management philosophy. Even in a relatively small company like Zycon (1100 employees), completely undergoing this cultural change has been a big obstacle. Zycon's management has been attempting to implement TOC for three to four years, yet the company still encounters problems related to cost-world thinking. Clearly, an organization as large as AFLC, and one that is still governed strictly in accordance with cost accounting principles, will have even more problems overcoming this mind-set. *DISASTER*™ will definitely not work in environments where managers still rely on cost-based thinking nor will it work when policy constraints are present. The ability to use *DISASTER*™ will hinge upon whether AFLC can completely change its management philosophy.

Another question that always arises during discussions of applying TOC to military organizations is "what is the goal?" When one considers applications within AFLC depot operations, the answer to this question is more straightforward. AFLC is not in business to make money, however, as evidenced by the command's current emphasis on performance to budget, they clearly strive to minimize cost; therefore, while the goal of the overall command is to maintain maximum readiness, operating at minimum cost must be at least a secondary goal.

Another obstacle AFLC will need to overcome to use *DISASTER*™ is the current segmentation of data between various systems and locations. While all of the information has likely been collected for DMMIS, it is not clear whether all the required data is in the same computer format or if it can be easily assembled in one location. Likely, as was the case with Zycon, given the appropriate level of attention and time, this problem can easily be overcome.

The research also highlighted the need for an accurate job tracking system. For *DISASTER™* to properly schedule operations, accurate information concerning the status of jobs through the entire productive system must be maintained. Zycon's original tracking system excluded the up-front engineering process. For *DISASTER™* to properly manage interactivity between all system constraints (and truly maximize throughput), an accurate job tracking system must be in place for the entire productive system.

This idea is very important to application of *DISASTER™* to AFLC operations. In the past, instead of attempting to maximize the output of the system as a whole, each directorate has typically attempted to maximize its own performance. Apparently, each center normally assumes that maximizing the performance of each of the individual directorates will lead to maximizing the production of the center. Unfortunately, as indicated by the research, this assumption is inappropriate. For any AFLC planning and control system, including DMMIS, to be completely effective, each air logistics center must be regarded as the productive system, not the individual directorates (or product directorates). AFLC will not be able to maximize their throughput until they begin viewing operations from the perspective of the entire productive system, i.e., at least the air logistics center. Optimizing the performance of each individual organization with the center (the directorates/product directorates) will definitely not optimize the operation of the center as a whole.

Recommendations for Future Research

This research has only explored the "tip of the iceberg." As noted, *DISASTER™* was only recently released, and two more phases of the

software are in development. *DISASTER*™ has yet to be applied; therefore no information exists regarding limitations and problems that have resulted from its use. While the use of *DISASTER*™ appears worthwhile, the reader should be cautioned that this work is based primarily upon theory: additional study is definitely required.

Several areas are suggested for follow-on research. First, the *DISASTER*™ schedule block definitely needs to be applied, on a small-scale basis, in a real-world, Air Force organization. Given AFLC's current interest in the theory of constraints, a depot maintenance organization would be the logical place to test the system. In addition, studies similar to this thesis, followed by real-world applications, should be undertaken to examine software for the control and the what-if phases of *DISASTER*™ as soon as they are released.

Bibliography

Air Force Logistics Command. *Depot Maintenance Management Information System (DMMIS).* Student Training Handbook, SM-ALC/MA, McClellan AFB CA, January 1990a.

-----. *FY89 Depot Maintenance Annual Report.* AFLCP 66-65, Wright Patterson AFB OH, HQ AFLC/MA, January 1990b.

-----. *Logistics Management Systems.* HQ AFLC, LMSC/SZ, Wright Patterson AFB OH, 1989a.

-----. *Quarterly Major Automated Information System Status Report RCS: DD-COMP(Q)1799.* AFLC LMSC/SQ, Wright Patterson AFB, OH, December 1989b.

-----. *Justification for the Development of the Depot Maintenance Management Information System.* DAR LOG-MAS-G81-090, Amendment D, AFLC/LMSC/SQPM, Wright Patterson AFB OH, June 1988.

Anderson, John C., Roger G. Schroeder, Sharon E. Tupey, and Edna M. White. "Material Requirements Planning Systems: The State of the Art," *Production and Inventory Management*, 51-67 (Fourth Quarter 1982).

APICS Dictionary (Sixth Edition). Falls Church VA: American Production and Inventory Control Society, 1984.

Avraham Y. Goldratt Institute. *CALENDAR User's Guide* (Version 1). Documentation Accompanying Disaster, New Haven CT: Avraham Y. Goldratt Institute, 1990a.

-----. *Jump Start* (Version 1). Documentation Accompanying Disaster, New Haven CT: Avraham Y. Goldratt Institute, 1990b.

-----. *NETGEN User's Guide* (Version 1). Documentation Accompanying Disaster, New Haven CT: Avraham Y. Goldratt Institute, 1990c.

-----. *Read Me First!* (Version 1). Documentation Accompanying Disaster, New Haven CT: Avraham Y. Goldratt Institute, 1990d.

Berger, Gus. "Ten Ways MRP Can Defeat You," *Conference Proceedings. APICS*, 1987, pp. 240-43).

Borg, Walter R. and Meredith D. Gall. *Educational Research: an Introduction* (Second Edition). New York: David McKay Company, Inc., 1971.

Bylinski, Gene. " An Efficiency Guru with a Brown Box," *Fortune. 108:* 120-132 (September 1983).

Chase, Richard B. and Nicholas J. Aquilano. *Production and Operations Management* (Fifth Edition). Boston: Irwin, 1989.

Clark, Steven J. and others. "How to Evaluate Your Material Requirements Planning System." *Production and Inventory Management*. 15-32. (Third Quarter 1982).

Cox, James F. and others. "Auditing an MRP System." *Academy of Management Journal*. 386-402 (Second Quarter 1981).

Cusumano, Michael A. "Manufacturing Innovation: Lessons from the Japanese Auto Industry," *Sloan Management Review*, *30:* 29-39 (Fall 1988).

Demmy, Steven W. and Anthony J. Giambrone. "MRP II in AFLC Maintenance Planning and Control." *Air Force Journal of Logistics*, *14*, No. 4. 7-11, (Fall 1990).

Drucker, Peter F. "The Emerging Theory of Manufacturing," *Harvard Business Review:* 94-102 (May-June 1990).

Dunning, John. Senior Vice President for Marketing. Personal Interview. Zycon Corporation, Santa Clara CA, 14 Jun 1991.

Edwards, James B. and Julie A. Heard. "Is Cost Accounting the No. 1 Enemy of Productivity?" *Management Accounting*, *65:* 44-49 (June 1984).

Faulkner, John M. *The Impact of MRP II Implementation at Ogden Air Logistics Center on Future Air Force Logistics Command Implementations*. MS thesis, AFIT/GLM/LSM/89S-21. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright Patterson AFB OH, September 1989 (AD-A215-746).

Finnern, Michael L. *A Case Study of the Implementation of Manufacturing Resource Planning at the Ogden Air Logistics Center*. MS thesis, AFIT/GLM/LSM/88S-18. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright Patterson AFB OH, September 1988 (AD-A201565).

Fox, Robert E. "Main Bottleneck on the Factory Floor?" *Management Review*, *73:* 55-61 (November 1984).

Franzino, Steve. Innerlayer Department Manager. Personal Interview. Zycon Corporation, Santa Clara CA, 14 Jun 1991.

Gishi, Alan. Materials and Production Control Manager. Personal Interview. Zycon Corporation, Santa Clara CA, 13 Jun 1991.

Goldratt, Eliyahu M. *The Haystack Syndrome: Sifting Information out of the Data Ocean*. New York: North River Press, Inc., 1990a.

-----. *What is this Thing Called Theory of Constraints and How Should it Be Implemented?* New York: North River Press, Inc., 1990b.

----- and Robert E. Fox. *The Race*. New York: North River Press, 1986a.

----- and Jeff Cox. *The Goal: A Process of Ongoing Improvement*. New York: North River Press, Inc., 1986b.

Hay, Edward J. *The Just-In-Time Breakthrough*. New York: John Wiley and Sons, 1988.

Jo, Larry. Director of Engineering Services. Personal Interview. Zycon Corporation, Santa Clara CA, 13 Jun 1991.

Kanet, John J. "MRP 96: Time to Rethink Manufacturing Logistics." *Production and Inventory Management Journal:* 57-61 (Second Quarter 1988).

Kilmer, Carleton F. and Richard J. Golden. "Manufacturing Resource Planning is Coming of Age in Defense." *Defense Management Journal. 22:* 18-23 (First Quarter 1986).

Lang, Gerhard, and George D. Heiss. *A Practical Guide to Research Methods* (Third Edition). Lanham MD: University Press of America, 1984.

Madarieta, Tony. CUPOSIT Manager. Personal Interview. Zycon Corporation, Santa Clara CA, 13 Jun 1991.

Melton, Marcus. "OPT--Fantasy or Breakthrough," *Production and Inventory Management*. 127: 13-21 (Second Quarter 1986).

Metzger, Anthony F. "MRP II (Manufacturing Resource Planning): Practical Theory and Implementation," *Production and Inventory Management. 25:* 49-61 (Second Quarter 1984).

Powell, Bill. "Boosting Shop-Floor Productivity by Breaking all the Rules," *Business Week. 2840:* 100-104 (November 1984).

Sandman, William E. *How to Win Productivity in Manufacturing*. Dresher, Pa: Yellow Book of Pennsylvania, 1980.

Schragenheim, Eli and Boaz Ronen. *The Drum Buffer Rope Shop Floor Control: An Application of the Theory of Constraints*. Working Paper number 19/89. Israel Institute of Business Research, May 1989a.

-----. *Buffer Management: A Diagnostic Tool for Production Control*. Working Paper number 22/89. Israel Institute of Business Research, June 1989b.

Shoemaker, Larry. Senior Vice President for Finance. Notes and Figures Provided During Personal Interview. Zycon Corporation, Santa Clara CA, 13 Jun 1991a.

-----. Senior Vice President for Finance. Personal Interview. Zycon Corporation, Santa Clara CA, 13 June 1991b.

-----. Senior Vice President for Finance. "Theory of Constraints Weekly Status Report," Report to Zycon Jonah Group. Zycon Corporation, Santa Clara CA, 6 June 1991c.

-----. Senior Vice President for Finance. "Prepared for the 90's: A Process of Ongoing Improvement." Copy of Briefing on Zycon TOC History. Zycon Corporation, Santa Clara CA, August 1990.

Teresko, John. "MRP II: A Strategic Tool for Survival," *Industry Week, 226:* 38-42 (September 1986).

Trigger, Lewis S. *Investigating the Application of the Theory of Constraints to the Scheduling Environment of the IAF's Depots.* MS Thesis, AFIT/GLM/LSM/90D-61. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright Patterson AFB OH, September 1990.

Umble, Michael M. and M. L. Srikanth. *Synchronous Manufacturing.* Cincinnati: South-Western Publishing Co., 1990.

Vollman, Thomas E., William L. Berry, and D. Clay Whybark. *Manufacturing Planning and Control Systems.* Homewood IL: Irwin, 1984.

Wong, Ken. Assistant Innerlayer Department Manager. Personal Interview. Zycon Corporation, Santa Clara CA, 13 Jun 1991.

Wright, Ron. "Disaster Work Order Model." Notes and Chart Provided During Personal Interview. Zycon Corporation, Santa Clara CA, 14 June 1991a.

-----. MIS Department Manager. Personal Interview. Zycon Corporation, Santa Clara CA, 14 Jun 1991b.

Yin, Robert K. *Case Study Research: Design and Methods.* Beverly Hills: Sage Publications, 1984.

Zikmund, William G. *Business Research Methods* (Second Edition). Chicago: The Dryden Press, 1988.

Zycon Corporation. "How to Make a Printed Circuit Board." Chart Provided During Personal Interview with Mr. Ken Wong. Zycon Corporation, Santa Clara CA, 13 June 1991.

Vita

Captain Jeff L. Severs was born on 17 February 1963 in Morgantown, West Virginia. He graduated from Warner Robins High School in 1981 and attended the University of Georgia from 1981 to 1985, earning a Bachelors' in Business Administration degree (majoring in management information systems) in August, 1985. He also received a commission in the USAF through the Air Force Reserve Officers Training Corps in August of 1985. Upon entering active duty, he attended the Communications Officer Core Course at Keesler Air Force Base, Mississippi, from October, 1985 to January, 1986. From January, 1986 to April, 1990, Captain Severs was assigned to McClellan Air Force Base, Sacramento Air Logistics Center (SM-ALC), AFLC, and worked in a variety of positions. Within the Directorate of Materiel Management he performed duties as an acquisition program manager for electronic warfare equipment, and within the Directorate of Maintenance, he worked as Deputy Chief, Scheduling and Inventory Control Branch within the Communications, Electronics and Space Maintenance Division. His final position at SM-ALC was as Executive Officer to the Directorate of Maintenance. Captain Severs was assigned as a student to the Air Force Institute of Technology, School of Systems and Logistics, in April of 1990.

Permanent Address: 1901 Northside Road
Perry, Georgia 31069

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1991

3. REPORT TYPE AND DATES COVERED
Masters Thesis

4. TITLE AND SUBTITLE
THE DISASTER™ SCHEDULING SYSTEM:
A REVIEW AND CASE ANALYSIS

5. FUNDING NUMBERS

6. AUTHOR(S)
Jeff L. Severs, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Wright Patterson AFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GLM/LSM/91S-56

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This research examined the capabilities of a new scheduling system, called DISASTER, based on the theory of constraints. Use of this system may be beneficial to the Department of Defense, particularly AFLC maintenance organizations. The research reviews literature pertaining to manufacturing planning and control systems, including material requirements planning (MRP), manufacturing resource planning (MRP II), AFLC's Depot Maintenance Management Information System (DMMIS), just-in-time (JIT) manufacturing, and the theory of constraints drum-buffer-rope scheduling. The research then examined the logic and operation of the DISASTER system. Next the research examined the implementation of DISASTER by a commercial printed circuit board manufacturer.

The research indicates that a growing number of experts now believe that MRP systems do not provide adequate shop-floor scheduling and control. While AFLC's DMMIS will certainly be an improvement over the command's present mode of operation (i.e., 1950s and 1960s vintage batch processing systems), this research suggests that other more recently developed alternatives, such as DISASTER, may be even more advantageous.

14. SUBJECT TERMS
Theory of Constraints, Synchronous Manufacturing, Depot Scheduling, Manufacturing Planning and Control

15. NUMBER OF PAGES
188

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT Control Number AFIT/GLM/LSM/91S-56

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

   Man Years ________________ $____________________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

Name and Grade ____________________ Organization ____________________

Position or Title ____________________ Address ____________________